3. REMARKS

Applicants wish to thank the Examiner for having done such a thorough job not only of reviewing the claims and studying the prior art but also of studying the drawings and the specification of this lengthy patent application, pointing out defects that truly do need to be addressed. Also, applicants wish to thank the Examiner for having reviewed the claims of several other related applications that are continuations or continuations-in-parts of the present application. There are seven such related applications, and for the Examiner's convenience, copies of their claims are set forth in an Appendix B of this response.

After carefully studying the Examiner's rejection, the specification and drawings, the claims filed with this application, and the claims in other applications which the Examiner has pointed out in his double-patenting rejection; and after preparing corrections to the specification and drawings; applicants have elected to cancel all of the claims 1 to 23 originally filed with this application and to prepare three entirely new sets of claims (method, system, and "means for" system) for consideration by the Examiner. These newly-submitted claims are directed in general toward the inventions claimed by the cancelled claims 11 and 18. They have been carefully crafted with all the Examiner's objections in mind in a good faith effort to seek out an allowable set of claims for this application.

Applicants respectfully request that all of the foregoing amendments be made prior to examination of the present application. Applicants also respectfully request reconsideration of the present application, and its newly-submitted claims, in view of the foregoing amendments and in view of the reasons set forth below.

A. Corrections Made To The Drawings

In the drawings, formal drawings have been submitted, revised so that the figure identification labels have the same style on all of the drawing sheets. In addition, Figures 2, 16, and 23 have been amended as is indicated in the annotated copies of the previously-submitted drawing sheets, which appear in Appendix A. All of the Examiner's objections to the drawings

have been addressed and corrected, either by these corrections or by the addition of reference numbers to the specification text.

In particular, the box in Figure 16 that lacked a reference number now has the reference number 1606, which appears in paragraph 173. In Figure 23, the number 2210 and its connecting line have been corrected. And the description now specifically mentions the reference number 1406 (in Figure 14) in amended paragraph 158, to which a sentence has been added containing the reference number 1406 as well as text substantially derived from the descriptive label applied the box 1406 in Figure 14. The label in the box 1406 reads: "MANAGER OF ANALYZER ISSUE LOOKUP TABLES RELATING ANALYZER TO ISSUES" while the sentence newly added to paragraph 158 reads: "A manager 1406 may also be provided that manages analyzer issue lookup tables which relate analyzers to specific issues." Since this newly-added language was taken from Figure 14 with very little change, it does not constitute new matter.

B. Corrections Made To The Specification

In the specification, paragraphs 49, 81, 100, 122, 126, 127, 130, 131, 132, 133, 134, 143, 150, 158, and 201 have been amended. All of the Examiner's objections to the specification have been addressed, and other corrections have been made as well.

The Examiner objected, with respect to paragraph 49, that: "The analyzer harness on page 8 is not in FIG. 21. Is should be identified as item 806 in FIG. 2." With all due respect, this is not correct – the analyzer harness 806 does appear in Figure 21, and it is correctly labeled as item 806 in that figure. In Figure 21, the analyzer harness 806 is indicated by a dashed line, correctly labeled 806, that encloses most of the other elements in Figure 21. The elements 2102, 2104, 2106, 2110, 2112, 2114, 2116, 2124, and 2126 all fall within, and form parts of, the analyzer harness 806. In response to this objection, applicants have amended paragraph 48 to clarify that the analyzer harness 806 appears in all three of the drawing Figures 2, 8, and 21, and that its details appear in Figure 21. In addition, applicants have also amended the preceding paragraph 47 to indicate more clearly that the drawing Figure 21 is actually a detailed drawing of

the analyzer harness 806 and of its elements. Applicants hope that this explanation, together with these clarifying amendments, will satisfy the Examiner fully.

The Examiner also indicated that "The link between the analyzer server 800 and enterprise 300 on page 13 [0081] is not visible." Actually, this linkage is shown in Figure 2, but the linkage is not a direct linkage – it passes through and includes a tracker database 106, as can be seen in Figure 2. In response to this objection, applicants have corrected Figure 2 (by adding the reference number 323 to the line connecting the elements 104 and 106) and has also corrected paragraph 81 of the specification (as indicated below) to clarify the description, without adding any new matter beyond what is clearly shown in Figure 2. As corrected, the drawing and this paragraph make it clear that:

> **[0081]** .... The analyzer server 800 is linked to the enterprise 300 via an ISDN line or some other form of wide area network 323 and by a tracker database 106 within an HAO server 105. .... Files of configuration information generated by the collectors 104 of the enterprise 300 are saved in the [[a]] tracker database 106. These files are retrieved from the tracker database 106 by an analyzer harness 806 within the analyzer server 800, as is described below.

In compliance with the Examiner's requirement, Applicants have found and corrected all incorrect references to "auditor 813" and to "content expert 812".

The one remaining correction made to paragraph 158 was made in response to one of the Examiner's objections to the drawings, namely, his objection that the reference number 1406 was not referred to at all in the description. (This correction was explained above.)

C. New Claims Presented

All of the claims 1 to 23 in the application as filed have been cancelled. New claims 24 to 42 are presented for the Examiner's consideration. Included are three sets of claims:

Independent computerized method claim 24, accompanied by dependant computerized method claims 25 to 29 and by a narrow, independent computerized method claim 30;

Independent computerized system claim 31, accompanied by dependant computerized system claims 32 to 36 and by a narrow, independent computerized system claim 37; and

Independent "means for" computerized system claim 38, accompanied by dependant "means for" computerized system claims 39 to 41 and by a narrow, independent "means for" computerized system claim 42.

D. Rejection of the Claims under 35 USC Section 112

The Examiner has rejected all of the claims as failing to comply with the enablement requirement, stating: "The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art ... to make and/or to use the invention." (Page 5, Office communication mailed 11/19/03)

Applicants respectfully request reconsideration of this grounds for rejection. The present invention, as described in the specification and drawings and as now claimed in the newly-submitted claims, is a system and method for monitoring automatically the hardware and software configurations of large numbers of networked computers (personal computers or servers or both) in enterprises. In particular, the present invention provides a simple and practical way in which very simple computer programs, called "analyzers" in the specification and in the claims, and written to analyze the configuration of just one single computer, can without the need for any more programming have their applicability greatly expanded so that they may be automatically applied to hundreds or even thousands of computers located at many different sites. Accordingly, the present invention reduces the very difficult task of writing a highly sophisticated data gathering, data testing, and report generating program that can monitor the

configurations of thousands of different computers at many sites to the much simpler task of writing very simple "analyzer" programs for each data gathering operation or test on the simplifying assumption that only one computer is being tested (when in reality many computers will be tested without the need for further programming).

As to practical applications of the invention, throughout the patent specification and its appendices, an exemplary analyzer is described that checks the disk space available on each mounted local hard disk drive of a computer, flagging as an issue any hard disk drive that has very little room left for additional data. This analyzer's executable code is presented in Appendix B of the application, and it contains only 53 lines of executable program code. There is no code in this program that defines how the program is to retrieve data from thousands of different computers. This code in Appendix B of the application simply analyzes some of the configuration data that is taken from a single computer. Yet, when this analyzer is applied in accordance with the teachings of the present invention and as claimed in the newly-submitted claims, this simple program is able to analyze the configuration of hundreds or even thousands of computers fully automatically, testing all of their hard drives and reporting any issues requiring the attention of the network manager, without the need for any additional programming.

The specification goes on to describe 25 additional practical applications of analyzers. Here are a few examples:

Paragraph 203 describes an analyzer which checks to see that there are both a primary and an alternate boot disk for every computer, and to insure that the two boot disks are not installed on the same hardware path. An issue is flagged if there is only one boot disk or if both are installed on the same hardware path.

Paragraph 212 describes an analyzer which tests memory by checking for the existence of single and double bit errors in a computer's hardware logs, among other things.

Paragraph 220 describes an analyzer which verifies the firmware revisions of all devices attached to a computer.

Paragraph 227 describes an analyzer which checks the internal processor hardware logs of a computer and reports on the status of the CPU fans, power supplies, etc..

In view of all this demonstration of the practical utility of the present invention, and in view of the fact that the claims closely track this practical utility, applicants respectfully request reconsideration of this grounds for rejection and allowance of the claims presently in this application.

In addition, since this rejection under Section 112 discussed and the rejection under Section 101 discussed below appear to be very closely related, applicants submit that all the reasons presented below as to why the claims are allowable under Section 101 are equally applicable as reasons why the claims are allowable under Section 112 as applied here.

E. Rejection of the Claims under 35 USC Section 101

The Examiner has further rejected the claims under 35 U.S.C. Section 101 "because the claimed invention is not supported by either a credible asserted utility or a well established utility." Applicants respectfully request reconsideration of this grounds for rejection.

First, the utility of the present invention was pointed out above in Section D of this response, in response to the Examiner's rejection under 35 U.S.C. Section 112. Applicants reassert all of Section D in response to the Examiner's Section 101 rejection. On that basis alone, reconsideration and allowance of the claims now before the Examiner is respectfully requested.

The Examiner first notes that certain claims "are not claimed to be practiced on a computer nor are they stored in a computer readable medium." Applicants submit that the newly-submitted claims are all directed to a "computerized method" or to a "computerized

system". The Examiner then notes that certain claims "are also not in the technological arts." This is clearly not true of the newly-submitted claims. Many of the claims also call for the storage of data in and the retrieval of data from databases (specifically, claims 25, 26, 29, 30, 32, 33, 36, 37, and 38-42). Reconsideration and allowance of the claims now before the Examiner is accordingly respectfully requested.

The Examiner next states that " ... none of ... [the claims] are limited to practical applications in the technological arts." The Examiner then quotes the following passage from applicable case law:

> ... [T]aking several abstract ideas and manipulating them together adds nothing to the basic equation. *In re Warmerdam, 33 F.3d 1354, 1360 (Fed. Cir. 1994)*

and then explains that "... Applicant[s'] 'configuration information' is just such an abstract idea."

Reconsideration of this grounds for rejection is respectfully requested. First of all, the phrase "configuration information" has been replaced in all of the newly-submitted claims with the expanded phrase "configuration information defining each computer's software configuration or hardware configuration or both."

Secondly, in *Warmerdam*, the Court found the invention to be unpatentable as defined by method claim 1, but found "machine" claim 4, which depended upon claim 1, to be patentable. Following the appeal, *Warmerdam* simply cancelled claim 4 and then amended just the preamble of claim 1 in the very minor way indicated below, and then his patent issued:

> 1. A machine having a memory which contains ~~method for generating~~ a data structure which represents the shape of a physical object in a position and/or motion control machine as a hierarchy of bubbles[[,]] generated by a method comprising the steps of:
>
> first locating the medial axis of the object and
> then creating a hierarchy of bubbles on the medial axis.

(U.S. Patent No. 6,089,742 issued to Thomas P. H. Warmerdam on Nov. 1, 1989)

Accordingly, all that is needed to overcome the *Warmerdam* Section 101 rejection is a simple recital of sufficient "computer" elements in the claim's preamble to limit the invention's applicability to use in a computer. The present applicants have done the same by including, in the preambles of all the newly-submitted claims, the limitations "A computerized method" or "A computerized system." Applicants submit that these claims are allowable under the law as defined in the *Warmerdam* decision. If this is still not sufficient, the Examiner is invited to call applicants' attorney to discuss what else might be needed to overcome this rejection.

Applicants submit further that patentability of applicants' invention is also fully supported by the case *State Street Bank and Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998), *cert. denied,* 525 U.S. 1093 (1999) which case the Examiner has also cited.

In *State Street*, the court found to be patentable a computer system that gathered many independent buy and sell orders relating to many independent mutual funds, combined these orders electronically, and then traded all the orders as if there were only one mutual fund, thereby gaining major benefits. From the market's point of view, there was just one mutual fund buying and selling stock; whereas from the shareholder's point of view, there were many separate mutual funds. The court in *State Street* held that this use of a computer to provide the appearance of a single mutual fund to the market and the appearance of many different mutual funds to the shareholders is a patentable use of a programmed computer.

Likewise, in the present invention, the computer system and its harness arrangement take a simple computer program, written by a programmer to test configuration data gathered from only one single computer, and apply that program over and over again to data gathered from hundreds or even thousands of different computers. From the programmer's point of view, there is only one simple computer that is being tested; but from the point of view of the enterprise manager, the computer system is testing hundreds or thousands of computers and generating audit reports identifying the issues arising on each individual computer. The present invention is thus of the same general type as the *State Stree* invention, namely an invention that presents the

appearance of only one computer being tested to the programmer (similar to *State Street* presenting the appearance of only one mutual fund to the market) and that presents the appearance of a machine that can test hundreds or thousands of computers to the enterprise manager (similar to *State Street* presenting the appearance of many mutual funds to the shareholders). Applicants submit that the *State Street* case thus fully supports the patentability of the present invention.

Warmerdam is distinguishable from the present invention in another important way: *Waarmerdam* is a case where a very specific mathematical equation was claimed and preempted – a specific set of equations that were actually set forth in *Warmerdam's* specification. The Court said: "the claim is mathematical in nature." (see 33 F.3d 1354 at 1360). The same may be said of the claims and inventions in *Allapat* (mathenatical calculations used to compute the brightness of each pixel in an oscilloscope display), *State Street* (mathematical equations used first to combine and later to separate security purchase and sale records and related tax records), and also of the claims and inventions in the early Supreme Court decisions *Benson* (mathematical formula for converting Binary Coded Decimal numbers into true binary numbers) and *Diehr* (Arrhenius's mathematical equation for calculating rubber cure times). The Supreme Court's rejection in *Benson* was summarized as follows:

> ... if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself. (*Gottshalk v. Benson,* 409 U.S. 63, at 71-71, 93 S.Ct. 253 at 257 (1972))

Contrary to the facts and claims in all of these cases, there is no specific mathematical equation or formula preempted by the present invention and its claims. It is true that the "analyzers," which form a claim element, are short computer programs, and they may be called algorithms and may perform mathematical operations. But some, if not most, of the analyzers disclosed in this application perform no significant mathematical steps, and no specific mathematical operations are called for by any of the claims presently before the Examiner. Anyone is free to use any "analyzer" computer program in any way they choose, except in conjunction with an "analyzer harness" and "collectors" and a "report generator," as claimed.

These other claim elements are also not mathematical equations. The analyzer harness 806, shown in Figure 21, is a device which picks up an analyzer program 2110, 2112, or 2114 (written in Java, C, or Perl) and then applies it to the testing of configuration information gathered from hundreds or even thousands of computers (this configuration information is found in the "collector reports" within the "tracker database" shown at 106 in Figure 21). The invention is thus a method or device that greatly magnifies the power and usefulness of very simple computer programs (called "analyzers") by using them to audit not one computer but many computers, thus greatly simplifying the process of designing automated systems for auditing many computers.

In view of the newly-submitted claims and the above remarks, applicants submit that the present invention, as now claimed, is directed to fully patentable subject matter under both Section 101 and also Section 112 of Title 35. Reconsideration and allowance of the claims is respectfully requested.

E. Double Patenting Rejection

In the double patenting rejection, the Examiner indicates that two of the claims now cancelled (claims 11 and 16) conflict with claims of two other applications of the same inventors and assigned to the same assignee. Applicants thank the Examiner for having drawn this matter of claim similarity to their attention.

The Examiner has not stated specifically that this rejection is "statutory double patenting" and the Examiner has not specifically cited Section 101 of Title 35 as the basis for such a rejection (see MPEP 804 II A). Applicants accordingly will presume that this rejection is based upon the judicially-created doctrine of "nonstatutory double patenting" (see MPEP 804 II B) which can be cured by the submission of a terminal disclaimer at any time (MPEP 804.02 II and 37 CFR 1.130(b)). At this time, with this application and both of the other two applications still pending in prosecution, applicants will presume in addition that this is a "provisional double patenting rejection," one that does not actually need to be addressed until only one application

containing each of the conflicting claims remains pending (see MPEP 822.01). Accordingly, the double patenting issue, if it is still present, does not have to be resolved at this time.

However, since the two claims 11 and 16 have both been cancelled from the present application for reasons unrelated to the double patenting rejection, there is no longer any conflict. The newly-submitted claims were drafted with the intent of avoiding any double-patenting problems.

There are actually seven other applications that are continuations or continuations-in-part of the present application. To assist the Examiner, applicants have added an Appendix B to this response. Appendix B contains copies of the first pages and also the pages containing the claims of all seven of these related applications, all of which have been published. Applicants invite the Examiner to review the published claims of these seven related applications and to compare them to the newly-submitted claims to see if any other double patenting problems, statutory or non-statutory, exist with respect to these newly-submitted claims.

F. Rejection of Claims 1 to 14 and 17 to 23 Under 35 U.S.C. Section 102(b)

All of the claims stand rejected in view of U.S. Patent No. 5,295,230 which issued on March 15, 1994 to Ching Y. Kung (hereinafter "Kung"). Reconsideration of this rejection, with respect to the newly-submitted claims, is respectfully requested.

1. Brief Summary of the Claimed Invention

The invention, as claimed here, is best illustrated in Figure 8 of the drawings (some claim elements appear in Figure 2).

The present invention, as defined in all the claims 24 to 42, teaches how one can start with a simple, easy-to-write computer program called an "analyzer" (a term defined in paragraph 55) written by, for example, an expert computer maintenance field engineer specifically to test the configuration of selected software or hardware components on just one single computer.

These "analyzers" are retrieved from an "analyzer database 804" and are then "harnessed" by means of an "analyzer harness 806" (a term defined in paragraph 56) to configuration information ("configuration" is defined in paragraph 59) gathered from many different computers (or "nodes" 302 and 304" in Figure 2 – "node" is defined in paragraph 70 to include computers) by "collectors 104" (Figure 2 - a term defined in paragraph 58) so that each analyzer is automatically and sequentially executed while harnessed to configuration information obtained from hundreds or even thousands of different computers. A "task definition" 814 or 816 determines which analyzers are selected and executed against information gathered from which specific computers. Any unusual or reportable events of interest to an auditor (for example, a maintenance engineer) are reported as "issue identifying reports." The term "issue" as used in the specification and claims has a special meaning and is defined in the specification as follows:

> Issue. An issue is any matter that may need to be investigated by or reported to the management of an enterprise. An analyzer performs tests upon configuration data gathered from the nodes of an enterprise by collectors. Those tests determine if there are any issues that need to be drawn to management's attention. (Specification, paragraph 67)

Claims 26 to 30, 33 to 37, and 39 to 42 include additional limitations that require the "identity of the computers whose configuration information was processed" to be included with at least some of the "issue identification reports," these computer identities constituting information which the very simple analyzer programs know nothing about and therefore cannot determine or report. This computer identification information is added automatically to the issue identification reports placed into an issues database 112 by the analyzer harness 806. Accordingly, each issue identification report is accompanied by the identity of the computer to which that particular issue relates.

And finally, claims 28 to 30, 35 to 37, and 40 to 42 all include additional limitations requiring the issue identification reports and (except for claim 40) the identity of the computers (in the issues database 112) to be fed into a report generator 206. The report generator 206 is also supplied with report templates (from the report templates and rules database 204) which are

also designated by the task definitions 814 or 816, so that highly customized and very readable audit reports are generated which identify and explain the issues and identify the computers upon which those issues arose. (The term "audit" is defined in paragraph 57.)

2. Brief Summary of the Kung Patent's Teachings

The Kung patent, in Figure 1, discloses a single "host computer 40" that is networked to the terminals 52, 64, 68, 72, 88, 90, 92, and 94 located at remote sites by means of: a front end processor 42; conventional modem pairs, such as the pairs 46 and 50; multi-drop (shared line) modems such as the modems 58, 62, 66, and 70; and multiplexed, multi-line modem pairs such as the modem pair 78 and 84. A "network manager 24" is connected by three RS 232 serial data cables (col. 10, lines 30 to 33) to the modems 46, 58, and 78 and also by a fourth RS 232 serial data cable to a multiplexer 74. The network manager 24 also communicates over the same three serial cables and through the same three central site modems 46, 58, and 78 with the five remote site modems 50, 62, 66, 70, and 84 using frequency division multiplexed secondary diagnostic channels that are entirely separate from the normal modem-to-modem communication channels (Col. 10 lines 33 to 37). Accordingly, these connections enable the network manager 24 to communicate with the all of the modems, both central and remote. (Col. 10, lines 41 to 43) Dial-up and X.25 networks are also suggested for use as part of this dedicated diagnostic network of modems (Col. 10, lines 58 to 66).

The network manager 24 also connects to a network manager terminal 33 and printer 34. It passes alarm message to the terminal and printer (col. 11, lines 5 to 7). The person managing the network may type on the terminal 33 requests to have diagnostic tests performed upon or data gathered from the remote network modems or multiplexers, and the results of these tests and requests are returned and displayed (col. 11, lines 7 to 15). The person managing the network may also view "data describing the network configuration" which is stored in a database manager 30 (col. 9, lines 51 to 53). In response to trouble, the person managing the network may send remedial instructions to an object (such as a modem) or switch in a redundant object or contact the telephone company for repairs. (col. 11, lines 10 to 15)

The "invention" in the Kung patent is a separate engineering workstation 12 containing a conventional expert system programming environment 16 in which has been developed into an expert network diagnostic system 10, the overview details of which appear in Figure 2 of the Kung patent. An RS 232 serial data cable connects the network manager 24 to this engineering workstation. In practical effect, Kung's expert network diagnostic system 10 may "manage" the network over the RS 232 cable 22 in the same way that a person can manage the network 128 working from the terminal 33. This is thus a classic example of an "expert system" that is intended to take over some or all of the monitoring, testing, and decision-making tasks of a human operator, with a "knowledge engineer" developing within the expert system programming environment 16 a set of rules (the expert information structure 11 shown in Figure 2) that can simulate human decision making, as is illustrated very broadly in Figure 3.

Of relevance to the present discussion, and with reference to Figure 2 of the Kung patent, configuration information sent in by the modems in the network is transferred through a configuration queue 113 to a network configuration module 108 which saves it in a network structure knowledge base 110. Alarms sent in by the modems in the network are transferred through an alarm queue 114, through an alarm filter 118, and onto a bulletin board 120. In response to such alarms, the inference engine 122, guided by the expert information structure 111, can call for tests to be performed or can request that data be gathered. In response, a network test manager 124 may generate test requests and may also generate information requests directed to specific modems, placing these requests into a request queue 115. These requests are eventually sent out over the diagnostic network to the modems, etc., and any responses are then returned through a response queue 116 to the network test manager 124 which then places the test results or the returned information onto the bulletin board 120 for further possible analysis by the inference engine 122.

3. The Claims Are Patentable Over Kung

Comparing the newly-submitted claims to the teachings of Kung, it can be seen that Kung does not teach the present invention.

First, all of the claims call for providing analyzers "comprising the executable program steps needed to compute, from selected configuration information gathered from a single computer, a report identifying at least one issue relating to the computer" (Independent Claims 24, 30, 31, 37, 38, and 42). There is no such teaching to be found in the Kung patent. Kung does not even teach gathering information from computers—he gathers information only from modems and the like. Kung does not teach providing a program written to process data from only one device when the intent is to process information gathered from many devices.

Secondly, all of the claims call for:

> "harnessing each of the ... listed analyzers to configuration information gathered from each of the ... listed computers" and then "processing the configuration information so harnessed under the guidance of each analyzer's executable program steps" (independent method claims 24 and 30);
>
> "an analyzer harness ... sequentially harnessing each of the ... listed analyzers to configuration information gathered from each of the ... listed computers and executing each analyzer's executable program steps upon the harnessed configuration information" (independent system claims 31 and 37); or
>
> "analyzer harness means ... for sequentially harnessing ... each of the ... listed analyzer means to configuration information gathered from each of the ... listed computers, for executing each analyzer mean's executable program steps upon the harnessed configuration information" (independent system claims 38 and 42).

Kung does not appear to teach this at all. Even in an expert system, additional explicit programming of some type is required to cause an expert rule to be applied to specific multiple sets of gathered information. In the present invention, the use of task definitions to control the operations of the analyzer harness makes such additional programming unnecessary and clearly distinguishes the claimed invention from the teachings of Kung.

Additionally, all of the claims call for a plurality of networked computers and for collectors that gather information defining how the hardware or software of the computers is configured. Kung does not disclose in his patent any computers networked together – only serial interconnections between one host computer and a plurality of terminals. The only "network" he

discloses is a "secondary diagnostic channel" (col. 10, lines 62 to 66) that is not capable of computer-to-computer communication – it is dedicated to the gathering of diagnostic information from modems and a multiplexer. There is no teaching in Kung that configuration information is to be gathered from anything other than modems, a multiplexer, and "other objects" (col. 9, lines 61 to 65). Kung does not teach gathering configuration information from computers over a network through the use of collectors.

The additional requirement in claims 26 to 30, 33 to 37, and 39 to 42 that the "identity of the computers whose configuration information was processed" be included within the "issue identification reports" generated whenever an analyzer is processed is a requirement not taught in the Kung patent. This aspect of the invention permits audit reports to identify each computer even though the analyzer programs are incapable of identifying the computers audited.

And finally, the additional requirement of the claims 28 to 30, 35 to 37, and 40 to 42 that the issue identification reports and (except for claim 50) the identity of the computers be fed into a report generator 206 which is also supplied with audit report templates (designated by the task definitions) is another requirement not taught at any point in the Kung patent. This aspect of the invention enables the production of readable audit reports, carefully tailored to the needs of specific audiences.

In view of this, reconsideration of this rejection and allowance of the claims is respectfully requested.

Rejection of Claims 15 and 16 Under 35 U.S.C. Section 103(a)

These two claims have been cancelled. In the present application, no other claims remain that include, as limitations, the required use of XML data structures in the design of an enterprise auditing system. Accordingly, this basis for rejection is not applicable to any of the newly-submitted claims. Reconsideration of this rejection is accordingly respectfully requested.

G. Conclusion

Applicants believe that the present application, as amended, is now in condition for allowance. Early and favorable reconsideration and allowance of this application, as amended, is respectfully requested.

The Examiner is invited to contact the undersigned by telephone if it is felt that a telephone interview would advance the prosecution of the present application.

The Commissioner is hereby authorized to charge any additional fees which may be required regarding this application under 37 C.F.R. §§ 1.16-1.17, or credit any overpayment, to Deposit Account No. 06-1450. Should no proper payment be enclosed herewith, as by a check being in the wrong amount, unsigned, post-dated, otherwise improper or informal or even entirely missing, the Commissioner is authorized to charge the unpaid amount to Deposit Account No. 06-1450. If any extensions of time are needed for timely acceptance of papers submitted herewith, Applicants hereby petition for such extension under 37 C.F.R. §1.136 and authorizes payment of any such extensions fees to Deposit Account No. 06-1450.

Respectfully submitted,

Date March 19, 2007

By Michael D. Rechtin

FOLEY & LARDNER LLP
321 North Clark Street
Chicago, Illinois 60610
Customer Number: 27433
Telephone: (312) 832-4596
Facsimile: (312) 832-4700

Michael D. Rechtin
Attorney for Applicants
Registration No. 30,128

Appendix A

Annotated Drawing Sheets Showing Changes

FIG. 2 ENTERPRISE LINKED TO ANALYZER AND REPORT GENERATOR

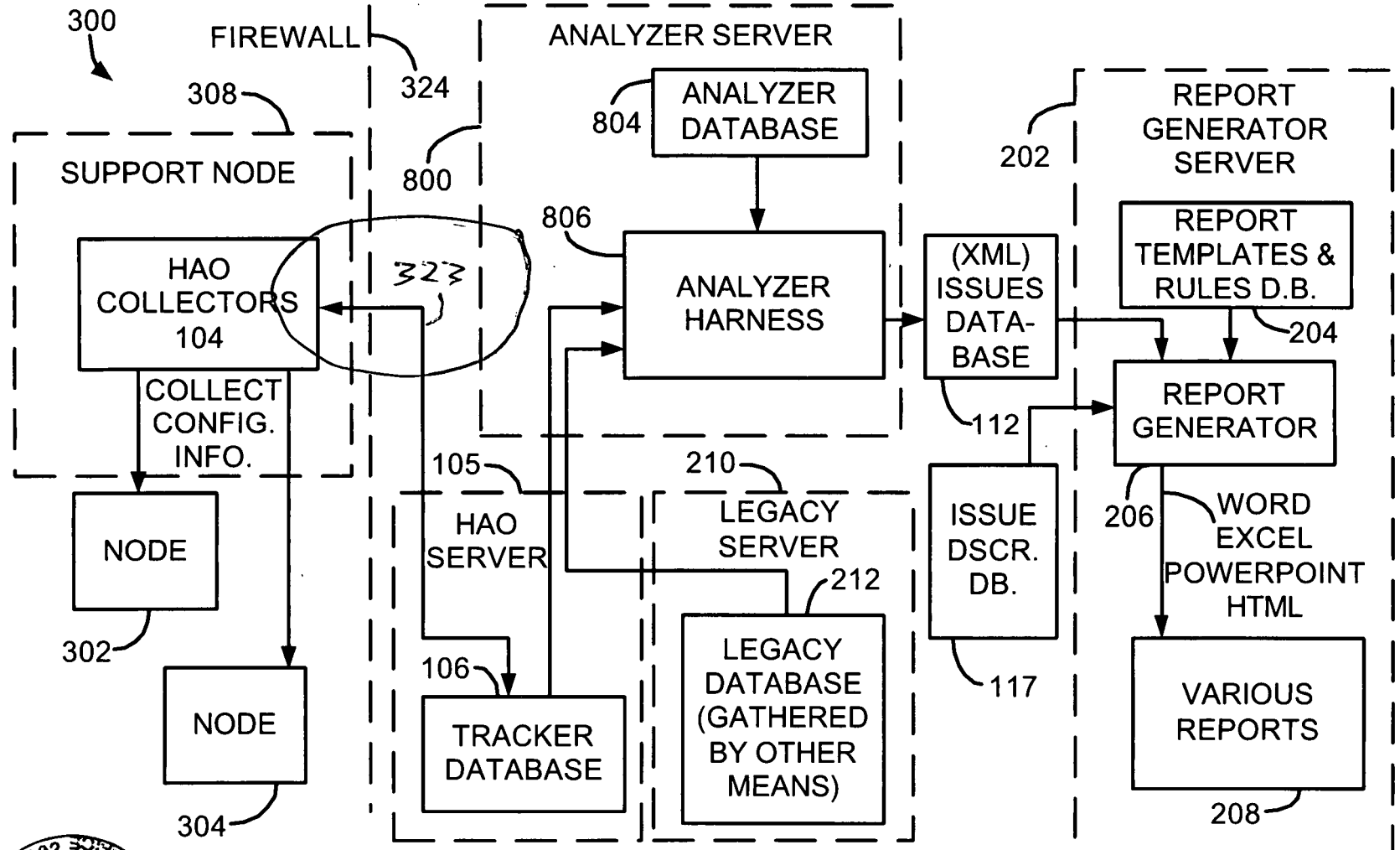


FIG. 16 REPORT TEMPLATE CREATION

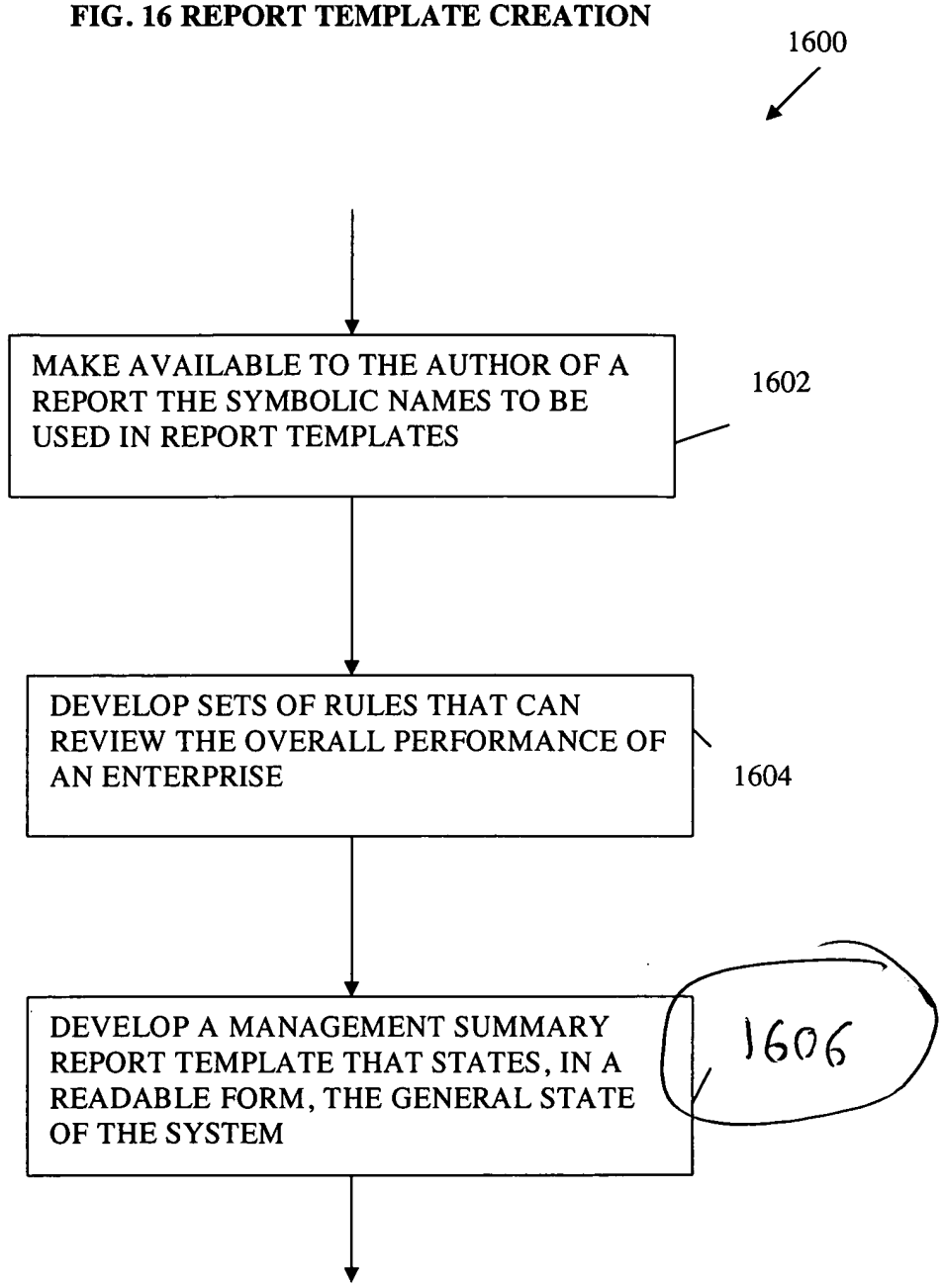
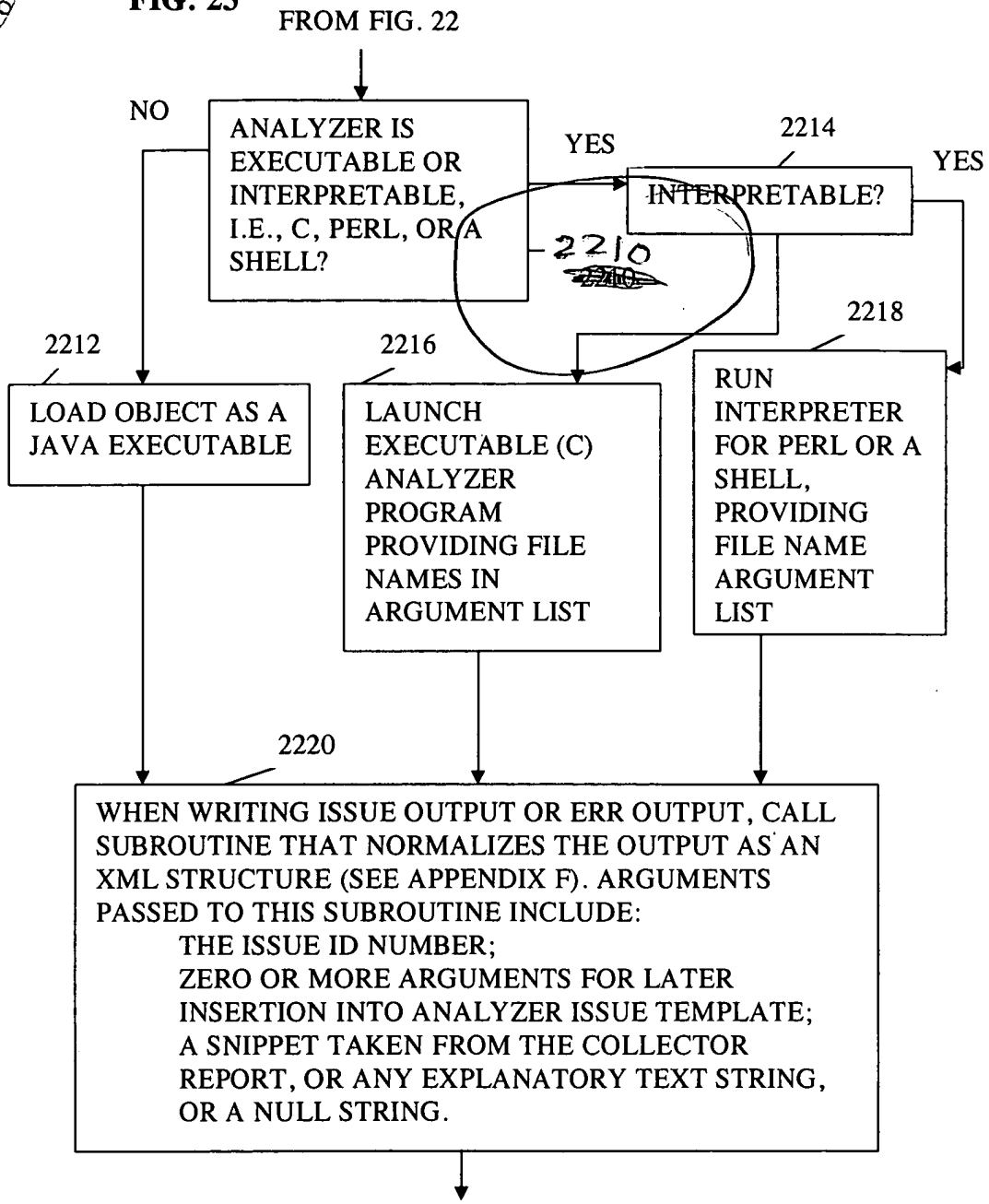


FIG. 23



Appendix B

First Pages and Claims of Related Continuation and Continuation-in-part Applications

US 20020169734A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2002/0169734 A1

Giel et al. (43) Pub. Date: Nov. 14, 2002

(54) APPARATUS AND METHOD FOR CAPTURING KNOWLEDGE THROUGH AN EXPERT INTERFACE

(76) Inventors: Peter Van Giel, Mesnil-Saint-Blaise (BE); Michael Joseph Brandon, Marietta, GA (US); William Paul Wedlake, Suwanee, GA (US); Peter A. Put, St. Albert (CA); Adam M. Carr, Fort Collins, CO (US); Paul Edward Holland, Fort Collins, CO (US); Mark W. McDowell, Ft Collins, CO (US)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: 10/152,556

(22) Filed: May 21, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/851,963, filed on May 10, 2001.

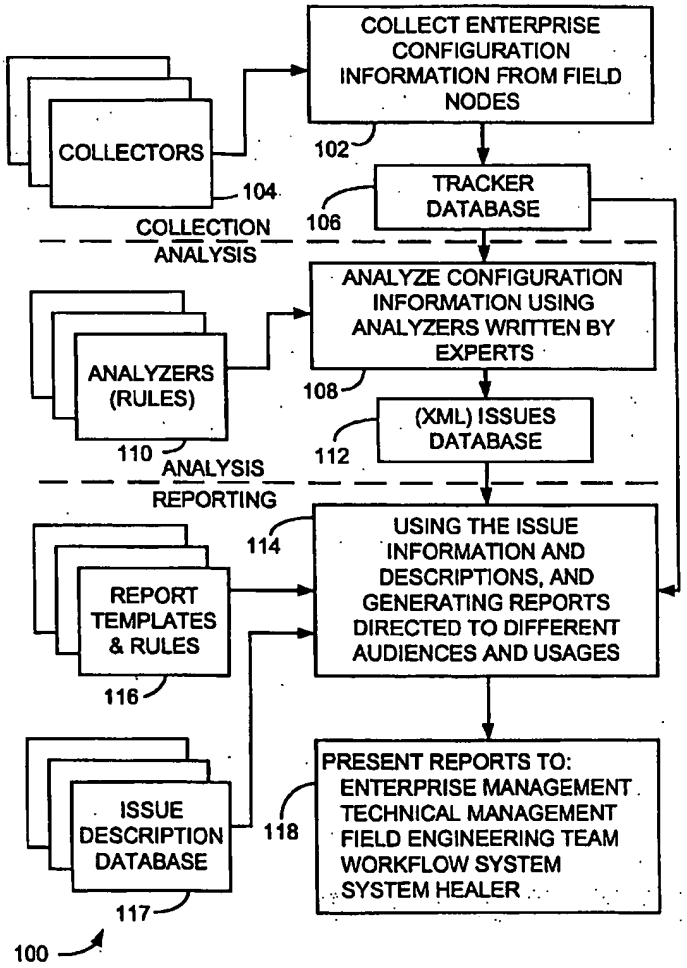
Publication Classification

(51) Int. Cl.[7] G06F 17/00

(52) U.S. Cl. 706/45; 707/100

(57) ABSTRACT

An apparatus and method is provided for auditing the configuration of an enterprise comprising the steps of: collecting information relating to the configuration of the enterprise, analyzing the configuration information based on expert knowledge; and providing the result of the analysis in the form of reports, and other results of the analyses.



What is claimed is:

1. A method of creating analyzers that can identify issues arising in the nodes of an enterprise having data collectors comprising the steps of:

for each of analyzer that is to be created,

creating analyzer code that accepts the output of one or more collector types, performs computations to detect the presence of issues, and outputs identifiers of issues which are present,

creating an issue template for each issue identifier defining an issue report, and

creating a descriptor of the analyzer identifying the collectors whose output the analyzer code requires as input; and

collecting the created elements of each analyzer into a database such that, when an analyzer is to be applied against the collector data for a node, the analyzer's descriptor is available to guide in the selection of collector data, the analyzer's code is available to guide in data analysis, and the issue templates are available to guide in the generation of any issue reports.

2. A method in accordance with claim 1 wherein the step of creating an issue template includes the step of providing for the inclusion in at east one issue report the identity of a particular node where the issue is present.

3. A method in accordance with claim 2 wherein the step of creating an issue template includes the step of creating within at least one issue template a position marker indicating where the identity of a particular node is to be placed within the issue report.

4. A method in accordance with claim 1 wherein at least some of the analyzer code is written in Java and/or an equivalent architecture independent language.

5. A method in accordance with claim 4 wherein at least some of the analyzer code is written in C and/or C++ and/or in an equivalent conventional language.

6. A method in accordance with claim 5 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

7. A method in accordance with claim 4 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

8. A method in accordance with claim 1 wherein at least some of the analyzer code is written in C and/or C++ and/or an equivalent conventional language.

9. A method in accordance with claim 7 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

10. A method in accordance with claim 1 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

11. A method in accordance with claim 1 wherein the output data is presented in XML or an equivalent data base compatible web page format.

12. A method in accordance with claim 1 wherein the descriptor is presented in XML or an equivalent data base compatible web page format.

13. A system for detecting issues arising on the nodes of an enterprise having data collectors comprising:

one or more analyzer programs that accept data from one or more collector types, that perform computations to detect the presence of issues, and that output identifiers of issues which are present;

for each issue identifier, an issue report template;

for each analyzer program, a descriptor identifying the collector types whose data the analyzer requires;

an analyzer harness that can exercise a set of one or more analyzer programs against data gathered from a different node or set of nodes during each such exercise, providing each analyzer program with collector data designated by the analyzer's descriptor and gathered from the node or set of nodes under examination; and

the analyzer harness responding to the output of issue identifiers by presenting reports formed in accordance with each issue identifier's issue report template.

14. A system in accordance with claim 13 wherein the analyzer harness augments at least some issue reports with the identity of the particular node or set of nodes where the analyzer has determined the identified issue is present.

15. A system in accordance with claim 14 wherein the issue report templates for at least some issue identifiers contain an indication of the position within an issue report where the analyzer harness is to insert the identity of the particular node or set of nodes where the analyzer has determined the identified issue is present.

16. A system in accordance with claim 13 wherein at least some of the analyzer code is written in Java and/or an equivalent architecture independent language.

17. A system in accordance with claim 16 wherein at least some of the analyzer code is written in C and/or C++ and/or in an equivalent conventional language.

18. A system in accordance with claim 17 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

19. A system in accordance with claim 16 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

20. A system in accordance with claim 13 wherein at least some of the analyzer code is written in C and/or C++ and/or an equivalent conventional language.

21. A system in accordance with claim 20 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

22. A system in accordance with claim 13 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

23. A system in accordance with claim 13 wherein the output data is presented in XML or an equivalent data-base compatible web page format.

24. A system in accordance with claim 13 wherein the descriptor is written in XML or an equivalent data-base compatible web page format.

25. A method of identifying the presence of issues through the analysis of output data provided by collectors monitoring the nodes of an enterprises, comprising the steps of:

providing analyzer programs that define the steps of accept data from one or more collector types, performing computations to detect the presence of issues, and that output identifiers of any issues which are present;

for each issue output identifier, providing an issue template that defines an issue report for that issue;

for each analyzer program, providing a descriptor identifying the collector types whose data the analyzer requires as its input; and

repeatedly exercising a set of the analyzer programs against output data gathered from different nodes or sets of nodes, providing each analyzer program with data from the types of collectors designated by the program's descriptor; and

when issue identifiers are generated, generating an issue report guided by the corresponding issue template.

26. A method in accordance with claim 25 further including the step of inserting the node identity of the node or set of nodes where an issue arose into at least some of generated issue reports.

27. A method in accordance with claim 26 which includes, in the step of providing the issue templates for at least some issues, the step of indicating within the issue template where the identity of the node or set of nodes under analysis may be inserted.

28. A method in accordance with claim 25 wherein at least some of the analyzer code is written in Java and/or an equivalent architecture independent language.

29. A method in accordance with claim 28 wherein at least some of the analyzer code is written in C and/or C++ and/or in an equivalent conventional language.

30. A method in accordance with claim 29 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

31. A method in accordance with claim 29 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

32. A method in accordance with claim 25 wherein at least some of the analyzer code is written in C and/or C++ and/or an equivalent conventional language.

33. A method in accordance with claim 32 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

34. A method in accordance with claim 21 wherein at least some of the analyzer code is written in perl and/or kshell and/or an equivalent operating system shell language.

35. A method in accordance with claim 21 wherein the output data is presented in XML or an equivalent data-base compatible web page format.

36. A method in accordance with claim 21 wherein the descriptors are presented in XML or an equivalent data-base compatible web page format.

37. A system for detecting issues arising on the nodes of an enterprise having data collector means for collecting enterprise state data comprising:

one or more analyzer means for accepting data from one or more collector means, for performing computations to detect the presence of issues, and for outputting identifiers of issues which are present;

for each issue identifier, issue report template means for defining the structure of an issue report;

for each analyzer means, descriptor means for identifying at least the type of collector means whose data the analyzer means requires;

analyzer harness means for exercising a set of one or more analyzer means against data gathered from a different node or set of nodes during each such exercise, for providing each analyzer means with data from collector means designated by the analyzer mean's descriptor means and gathered from the node or set of nodes under examination; and

the analyzer harness means including means for responding to the output of issue identifiers by presenting reports formed in accordance with each issue identifier's issue report template means.

* * * * *

US 20040015907A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0015907 A1

Giel et al. (43) Pub. Date: Jan. 22, 2004

(54) METHOD AND APPARATUS FOR AUTOMATIC SYSTEM CONFIGURATION ANALYSIS USING DESCRIPTORS OF ANALYZERS

(76) Inventors: Peter Van Giel, Mesnil-Saint-Blaise (BE); Michael Joseph Brandon, Marietta, GA (US); William Paul Wedlake, Suwanee, GA (US); Peter A. Put, St. Albert (CA); Adam M. Carr, Fort Collins, CO (US); Paul Edward Holland, Fort Collins, CO (US); Mark W. McDowell, Ft. Collins, CO (US)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: 10/151,802

(22) Filed: May 21, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/851,963, filed on May 10, 2001.

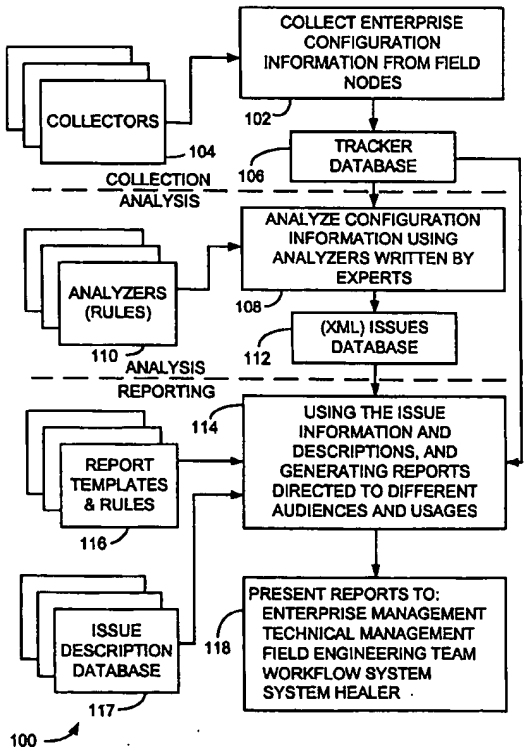
Publication Classification

(51) Int. Cl.[7] G06F 9/45

(52) U.S. Cl. 717/141; 717/142

(57) ABSTRACT

In a system that uses collectors to gather data indicating the status of nodes in an enterprise, analyzers are provided to analyze the collector data. Each analyzer includes a program and an accompanying XML analyzer descriptor. Each analyzer program is written in one of several programming language. Each descriptor identifys the types of collectors whose data the program accepts and also identifys the language in which the program is written. A framework exercises a set of one or more of these analyzer programs repeatedly, and during each such repetitive exercise of the set, presents the set of analyzers with collector output data gathered from a different specific set of one or more nodes. During each exercise of each analyzer program, the framework reviews the analyzer's descriptor and executes the program in a program execution environment appropriate to the language in which the program is written. The framework also notes the type of collectors whose data the program accepts and arranges to present this type of collector data to the program as a data input stream during program execution or interpretation.



What is claimed is:

1. In a system having collectors monitoring nodes, a method of analyzing collector data comprising the steps of:

providing one or more programs and descriptors, called analyzers, each program comprising a set of instructions written in a language chosen from a set of at least two languages and including instructions accepting and analyzing data from one or more collector types, and each descriptor comprising instructions written in XML or a comparable language identifying the types of collectors whose data the instructions accept and identifying the program language;

exercising a set of one or more of these analyzers repeatedly one or more times, and during each such exercise presenting the set of analyzers with collector output data gathered from a different specific set of one or more nodes; and

during each exercise of each analyzer, locating the analyzer's descriptor and program, exercising the program's instructions in accordance with the requirements of the program language identified by the descriptor, and when exercising instructions that accept and analyze collector data, exercising those instructions against data taken from collectors of the type identified by the descriptor.

2. A method in accordance with claim 1 wherein the descriptors identify the type of instructions the corresponding analyzer program contains (e.g.: text, interpretable, object, etc.) and also where the analyzer program is to be found.

3. A method in accordance with claim 1 wherein at least one of the languages in at least one analyzer program is Java and/or an equivalent architecture-independent interpretable language.

4. A method in accordance with claim 3 wherein at least some of the languages in at least one analyzer program is C or C++ or an equivalent compilable programming language.

5. A method in accordance with claim 4 wherein at least one of the languages in at least one analyzer program is perl or kshell or an equivalent operating system shell language.

6. A method in accordance with claim 3 wherein at least one of the languages in at least one analyzer program is perl or kshell or an equivalent operating system shell language.

7. A method in accordance with claim 1 wherein at least one of the languages in at least one analyzer program is C and/or C++ and/or an equivalent conventional language.

8. A method in accordance with claim 7 wherein at least one of the languages in at least one analyzer program is perl or kshell or an equivalent operating system shell language.

9. A method in accordance with claim 1 wherein at least one of the languages in at least one analyzer program is perl or kshell or an equivalent operating system shell language.

10. A method in accordance with claim 1 wherein at least one analyzer descriptor further identifies the author of the code, the code version, and contains a summary description of the operations performed by the corresponding analyzer.

11. A system that analyzes data gathered by collectors monitoring nodes, the system comprising:

one or more analyzers each comprising a program written in one of several different languages and accepting and analyzing data from one or more types of collectors, and a descriptor written in XML or a comparable language identifying the collector types whose data the analyzer program accepts and the language in which the analyzer program is written;

an analyzer framework that exercises a set of one or more analyzer programs repeatedly one or more times and that presents the set of analyzers, during each such repeated exercise of the set, with collector data gathered from a different specific set of one or more nodes;

the analyzer framework including plural program execution environments, at least one for each of the several different languages, and guided by the language identified in each analyzer's descriptor, the framework executing each analyzer's program within the proper execution environment; and

the analyzer framework including a collector data type selection mechanism that presents as a data input stream directed into the program execution environment of each analyzer program data obtained from collectors of the types identified by the corresponding analyzer descriptor.

12. A system in accordance with claim 11 wherein the analyzer descriptor further designates the program type (e.g.: text, interpretable, object, etc.) and where the analyzer program is to be found, and wherein the framework is further guided by this information in locating the analyzer program and in selecting an appropriate environment in which to exercising the analyzer program.

13. A system in accordance with claim 11 wherein at least one or more of the analyzer programs are written in Java or an equivalent architecture-independent interpretable language.

14. A system in accordance with claim 13 wherein at least one or more of the analyzer programs are written in C or C++ or an equivalent compilable language.

15. A system in accordance with claim 14 wherein at least one or more of the analyzer programs are written in perl or kshell or an equivalent operating system shell language.

16. A system in accordance with claim 13 wherein at least one or more of the analyzer programs are written in perl or kshell or an equivalent operating system shell language.

17. A system in accordance with claim 11 wherein at least one or more of the analyzer programs are written in C or C++ or an equivalent compilable language.

18. A system in accordance with claim 17 wherein at least one or more of the analyzer programs are written in perl or kshell or an equivalent operating system shell language.

19. A system in accordance with claim 11 wherein at least one or more of the analyzer programs are written in perl or kshell or an equivalent operating system shell language.

20. A system in accordance with claim 1 wherein at least one analyzer descriptor further identifies the author of the code, the code version, and includes a summary description of the operations performed by the corresponding analyzer.

21. A system that analyzes data gathered by collector means for monitoring nodes, the system comprising:

one or more analyzer means for accepting and analyzing data from one or more types of collectors means, each analyzer means comprising a program written in one of several different languages, and each analyzer means further comprising a descriptor written in XML or a comparable language identifying at least the collector

means types whose data the analyzer means's program accepts and the language in which the analyzer means's program is written;

analyzer framework means for exercising a set of one or more analyzer means repeatedly one or more times and for presenting the set of analyzer means, during each such repeated exercise of the set, with collector means data gathered from a different specific set of one or more nodes;

the analyzer framework means including means for exercising plural different types of analyzer means programs, at least one for each of the several different languages, and guided by the language identified in each analyzer's descriptor, the framework means including means for executing each analyzer mean's program within the proper execution environment; and

the analyzer framework means including data type selection means for selecting and presenting, as a data input stream directed into the program execution means, program data obtained from collector means of the types identified by the corresponding analyzer mean's descriptor.

* * * * *

US 20020169739A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2002/0169739 A1

Carr et al. (43) Pub. Date: Nov. 14, 2002

(54) METHOD AND APPARATUS FOR COMPOSITE ANALYSIS USING HIERARCHICALLY ORGANIZED DIRECTIVES

(76) Inventors: Adam Michael Carr, Fort Collins, CO (US); Paul Edward Holland, Fort Collins, CO (US); Mark William McDowell, Fort Collins, CO (US)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: 10/151,803

(22) Filed: May 21, 2002

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/851,963, filed on May 10, 2001.

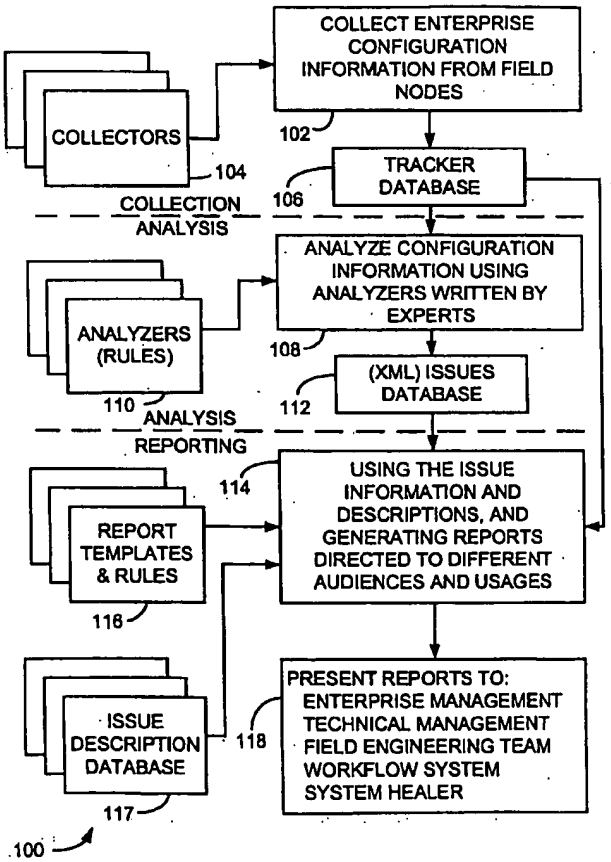
Publication Classification

(51) Int. Cl.[7] G06F 17/00

(52) U.S. Cl. 706/60

(57) ABSTRACT

A method and apparatus is provided for auditing the configuration of an enterprise using low-level analyzers that include a program and a descriptor of the type of collector data the program can accept, and high-level analyzers that are descriptors which call upon, or incorporate by reference, other analyzer descriptors. A framework, guided by one or more analyzer descriptors and by any descriptors those descriptors may call upon or incorporate by reference, repeatedly executes the set of programs associated with some of the analyzer descriptors against a different node or nodes of an enterprise during each such execution. When executing a program, the framework provides the program with the collector data of the type designated by the descriptor associated with the program.



What is claimed is:

1. In a system that analyzes data gathered by collectors monitoring nodes, a hierarchical method of defining and organizing the analysis tasks to be performed comprising the steps of:

performing, for each low-level analysis task, analysis steps that comprise accepting data from one or more collector types, analyzing the data, and generating reports, guided by an analyzer descriptor written in XML or a comparable format identifying the collector types whose data the analyzer steps are to process;

for each higher-level analysis task, of which there is at least one, performing one or more sets of analysis steps for either low-level or high-level analysis tasks, guided by an analyzer descriptor written in XML or a comparable format that calls upon or that incorporates by reference other analyzer descriptors; and

guided by one or more analyzer descriptors and, in the case of higher level analyzer descriptors, by the analyzer descriptors they call upon or incorporate by reference directly or indirectly, repeatedly performing the set of analysis steps corresponding to one or more analyzer descriptors one or more times against collector data gathered from a different node or set of nodes during each such exercise of the set of analysis steps, and during this process, presenting for acceptance and analysis during the performance of each data acceptance step data gathered from collectors of the type indicated by the guiding analyzer descriptors.

2. A method in accordance with claim 1 wherein the step of obtaining an analyzer descriptor includes obtaining one or more such descriptors that contain an analyzer name, description, and in the case of at least one higher-level analyzer descriptor a list of the other analyzer names whose descriptors it calls upon or incorporates by reference.

3. A method in accordance with claim 1 wherein the step of obtaining an analyzer descriptor for each higher-level task includes obtaining at least one analyzer descriptor for a higher-level task that calls upon or that incorporates by reference at least one other analyzer descriptor for another higher-level task.

4. A method in accordance with claim 3 wherein the step of obtaining an analyzer descriptor includes obtaining one or more such descriptors that contain an analyzer name, description, and in the case of at least one higher-level analyzer descriptor a list of other analyzer names whose descriptors it calls upon or incorporates by reference.

5. A method in accordance with claim 1 wherein the step of obtaining an analyzer descriptor for each higher-level task includes obtaining at least one analyzer descriptor for a higher-level task that calls upon or that incorporates by reference at least one other analyzer descriptor for another higher-level task which in turn calls upon or which incorporates by reference at least one other analyzer descriptor for another higher-level task.

6. A method in accordance with claim 5 wherein the step of obtaining an analyzer descriptor includes obtaining one or more such descriptors that contain an analyzer name, description, and in the case of at least one higher-level analyzer descriptor, a list of other analyzer names whose descriptors it calls upon or incorporate by reference.

7. A method in accordance with claim 1 wherein the step of obtaining an analyzer descriptor for each higher-level task includes obtaining at least one analyzer descriptor for a higher-level task that calls upon or that incorporates by reference at least one other analyzer descriptor for another higher-level task which in turn calls upon or which incorporates by reference at least one other analyzer descriptor for another higher-level task that calls upon or that incorporates by reference at least one other analyzer descriptor for another higher-level task.

8. A method in accordance with claim 7 wherein the step of obtaining an analyzer descriptor includes obtaining one or more such descriptors that contain an analyzer name, description, and in the case of at least one higher-level analyzer descriptor, a list of other analyzer names whose descriptors it calls upon or incorporate by reference.

9. A system that analyzes data gathered by one or more collectors monitoring one or more nodes, said system comprising:

for each low-level analysis task that the system performs, an analyzer program and an analyzer descriptor, the program containing instructions that accept data from one or more collector types, analyze the data, and generates reports, and the analyzer descriptor written in XML or a comparable format and identifying the collector types whose data the program's instructions can accept;

for hierarchically higher-level analysis tasks, of which there is at least one, at least one analyzer program, and an analyzer descriptor for each of the higher-level tasks written in XML or a comparable format that calls upon, or that incorporates by reference, other analyzer descriptors; and

a framework that is guided by a set of one or more analyzer descriptors and, in the case of hierarchically higher-level analyzer descriptors, by the analyzer descriptors they call upon or incorporate by reference directly or indirectly, to execute repeatedly one or more times the set of analyzer programs corresponding to some of these analyzer descriptors against collector data gathered from a different node or set of nodes during each such execution of the set of programs, and during each execution of each program, to present the program with only the type of collector data the instructions of the executing program are designed to accept, as indicated by the analyzer descriptor corresponding to the executing program.

10. A system in accordance with claim 9 wherein each analyzer descriptor contains an analyzer name, an analyzer description, and in the case of analyzer descriptors for higher-level analyzer tasks, a list of other analyzer names whose descriptors it calls upon or incorporates by reference.

11. A system in accordance with claim 9 including at least one or more higher-level analyzer descriptor that calls upon, or that incorporates by reference, another high-level analyzer descriptor.

12. A system in accordance with claim 9 wherein each analyzer descriptor contains an analyzer name, an analyzer description, and in the case of analyzer descriptors for higher-level analyzer tasks, a list of other analyzer names whose descriptors it calls upon or incorporates by reference.

13. A system in accordance with claim 9 including at least one or more higher-level analyzer descriptor that calls upon, or that incorporates by reference, another high-level ana-

lyzer descriptor which, in turn, also calls upon, or that incorporates by reference, yet another high-level analyzer descriptor.

14. A system in accordance with claim 9 wherein each analyzer descriptor contains an analyzer name, an analyzer description, and in the case of analyzer descriptors for higher-level analyzer tasks, a list of other analyzer names whose descriptors it calls upon or incorporates by reference.

15. A system in accordance with claim 9 including at least one or more higher-level analyzer descriptor that calls upon, or that incorporates by reference, another high-level analyzer descriptor which, in turn, also calls upon, or that incorporates by reference, yet another high-level analyzer descriptor that, in turn, also calls upon, or that incorporates by reference, another high-level analyzer descriptor.

16. A system in accordance with claim 9 wherein each analyzer descriptor contains an analyzer name, an analyzer description, and in the case of analyzer descriptors for higher-level analyzer tasks, a list of other analyzer names whose descriptors it calls upon or incorporates by reference.

17. A system that analyzes data gathered by one or more collector means for monitoring one or more nodes, said system comprising:

for each low-level analysis task that the system performs, an analyzer means for accepting data from one or more collector means, analyzing the data, and generating reports, and an analyzer descriptor written in XML or a comparable format and identifying the types of collector means whose data the program's instructions can accept;

for hierarchically higher-level analysis tasks, of which there is at least one, at least one analyzer means, and an analyzer descriptor written in XML or a comparable format for each of the higher-level analysis task that calls upon, or that incorporates by reference, other analyzer descriptors; and

framework means guided by a set of one or more analyzer descriptors and, in the case of hierarchically higher-level analyzer descriptors, by the analyzer descriptors they call upon or incorporate by reference directly or indirectly, for executing repeatedly one or more times the set of analyzer means corresponding to some of these analyzer descriptors against collector means data gathered from a different node or set of nodes during each such execution of the set of analyzer means, and during each execution of each analyzer means, for presenting the analyzer means with only the type of collector means data the analyzer means are designed to accept, as indicated by the analyzer descriptor corresponding to the executing analyzer means.

* * * * *

US 20040015908A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0015908 A1

Giel et al. (43) Pub. Date: Jan. 22, 2004

(54) APPARATUS AND METHOD FOR ANALYSIS DRIVEN ISSUE REPORT GENERATION

(76) Inventors: Peter Van Giel, Mensnil-Saint-Blaise (BE); Michael Joseph Brandon, Marietta, GA (US); William Paul Wedlake, Suwanee, GA (US); Peter A. Put, St. Albert (CA); Adam M. Carr, Fort Collins, CO (US); Paul Edward Holland, Fort Collins, CO (US); Mark W. McDowell, Ft Collins, CO (US)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: 10/152,214

(22) Filed: May 21, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/851,963, filed on May 10, 2001.

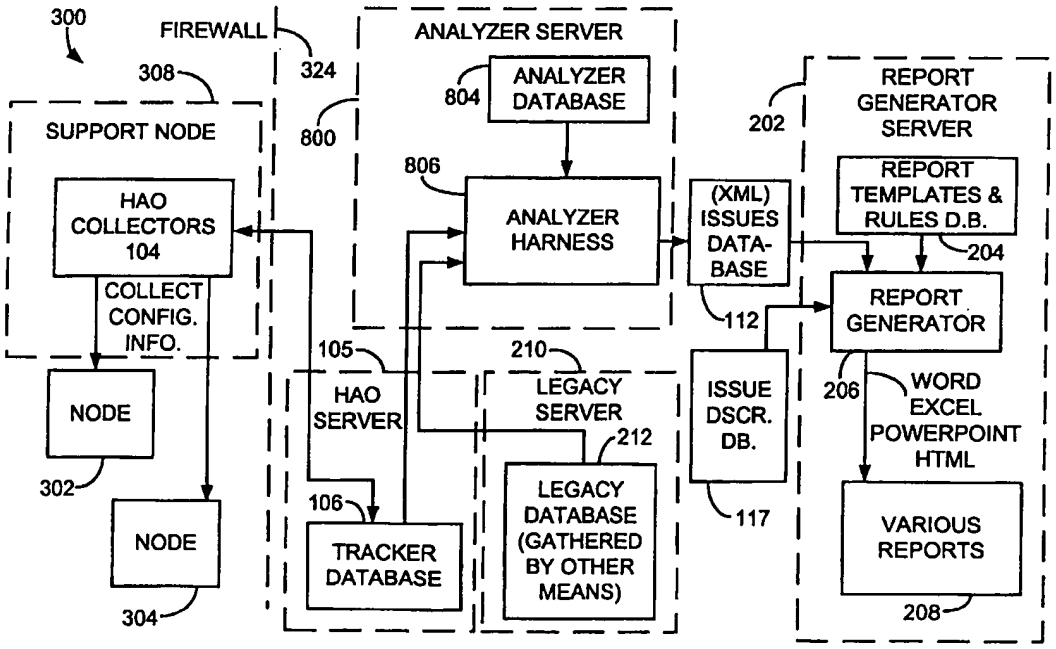
Publication Classification

(51) Int. Cl.[7] G06F 9/45

(52) U.S. Cl. 717/141; 717/144

(57) ABSTRACT

A computerized apparatus and method generates information descriptive of issues arising in a monitored set of nodes. The method comprises supplying one or more analyzer programs with node-specific information derived from one or more nodes. Each analyzer program is then caused to analyze at least some of the information provided from each node to detect the presence of one or more issues. When an issue is detected, the analyzer program is caused to generate issue identification information which is augmented with information identifying the node from which the node-specific information was derived. The augmented information for all the issues detected is presented in combined form as an issues database suitable for later use in report generation.



What is claimed is:

1. A computerized method for generating information descriptive of issues arising in a monitored set of nodes, said method comprising the steps of:

supplying one or more analyzer programs with node-specific information derived from one or more nodes;

causing each analyzer program to analyze at least some of the information provided from each node to detect the presence of one or more issues;

when an issue is detected, causing the analyzer program to generate issue identification information;

augmenting the issue identification information with information identifying the node from which the node-specific information was derived; and

presenting the augmented information for all the issues detected in combined form as an issues database suitable for later use in report generation.

2. A computerized method in accordance with claim 1 which further includes the step of augmenting the issue identification information with information descriptive of the issue identified.

3. A computerized method in accordance with claim 2 which further includes the step of having the analysis program generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

4. A computerized method in accordance with claim 2 which further includes the step of formulating the information descriptive of the issues identified as templates.

5. A computerized method in accordance with claim 4 which further includes the steps of having at least some of the analyzer programs generate, in addition to issue identification information, other issue-related information, including within at least some of the templates one or more spaceholders for issue-related information, and when such a template is used to augment issue identification information, replacing such spaceholders with this issue-related information.

6. A computerized method for generating information in accordance with claim 1 which further includes the step of formatting the augmented information for all the issues detected in combined form as an issues database formatted as an XML or similar document.

7. A computerized method for generating information in accordance with claim 6 which further includes the step of augmenting the issue identification information with information descriptive of the issue identified.

8. A computerized method in accordance with claim 7 which further includes the step of having the analysis program generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

9. A computerized method in accordance with claim 7 which further includes the step of formulating the information descriptive of the issues identified as templates.

10. A computerized method in accordance with claim 9 which further includes the steps of having at least some of the analyzer programs generate, in addition to issue identification information, other issue-related information, including within at least some of the templates one or more spaceholders for issue-related information, and when such a template is used to augment issue identification information, replacing such spaceholders with this issue-related information.

11. A computerized method for generating information in accordance with claim 6 which further includes the step of formatting the issue identification information as an XML or similarly-formatted document.

12. A computerized method for generating information in accordance with claim 11 which further includes the step of augmenting the issue identification information with information descriptive of the issue identified.

13. A computerized method in accordance with claim 12 which further includes the step of having the analysis program generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

14. A computerized method in accordance with claim 12 which further includes the step of formulating the information descriptive of the issues identified as templates.

15. A computerized method in accordance with claim 14 which further includes the steps of having at least some of the analyzer programs generate, in addition to issue identification information, other issue-related information, including within at least some of the templates one or more spaceholders for issue-related information, and when such a template is used to augment issue identification information, replacing such spaceholders with this issue-related information.

16. A computerized method for generating information in accordance with claim 1 which further includes the step of formatting the issue identification information as an XML or similarly-formatted document.

17. A computerized method for generating information in accordance with claim 16 which further includes the step of augmenting the issue identification information with information descriptive of the issue identified.

18. A computerized method in accordance with claim 17 which further includes the step of having the analysis program generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

19. A computerized method in accordance with claim 17 which further includes the step of formulating the information descriptive of the issues identified as templates.

20. A computerized method in accordance with claim 19 which further includes the steps of having at least some of the analyzer programs generate, in addition to issue identification information, other issue-related information, including within at least some of the templates one or more spaceholders for issue-related information, and when such a template is used to augment issue identification information, replacing such spaceholders with this issue-related information.

21. A system for generating information descriptive of issues arising in a monitored set of nodes comprising:

at least one analyzer program having provision to accept node-specific information, analyze the information for the presence of issues, and generate issue identification information when an issue is detected;

a harness that exercises one or more analyzer programs against node-specific information gathered from one or more nodes, presenting each analyzer program with at least some data from one of the nodes each time the program is exercised, and collecting any issue identification information generated by the programs, said harness augmenting the issue identification information generated with information identifying the node from which the node-specific information was obtained; and

an issues database containing the collected and augmented information for all the issues detected during such an exercising of the analyzer programs against node-specific information.

22. A system in accordance with claim 21 wherein the harness also augments the issue identification information with information descriptive of the issue identified.

23. A system in accordance with claim 22 wherein the analyzer programs also have provision to generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

24. A system in accordance with claim 22 wherein templates provide the information descriptive of the issues.

25. A system in accordance with claim 24 wherein the analyzer programs also have provision to generate, in addition to issue identification information, other issue-related information, including, wherein at least some of the templates contain one or more spaceholders for issue-related information, and wherein the harness inserts such issue-related information generated by an analyzer program into the template for an issue at positions indicated by the spaceholders.

26. a system in accordance with claim 21 wherein the issues database is formulated as an XML or similar document.

27. A system in accordance with claim 26 wherein the harness also augments the issue identification information with information descriptive of the issue identified.

28. A system in accordance with claim 27 wherein the analyzer programs also have provision to generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

29. A system in accordance with claim 27 wherein templates provide the information descriptive of the issues.

30. A system in accordance with claim 29 wherein the analyzer programs also have provision to generate, in addition to issue identification information, other issue-related information, including, wherein at least some of the templates contain one or more spaceholders for issue-related information, and wherein the harness inserts such issue-related information generated by an analyzer program into the template for an issue at positions indicated by the spaceholders.

31. A system in accordance with claim 26 wherein an output routine formulates the issue identification information generated by the analysis programs as an XML or similar document.

32. A system in accordance with claim 31 wherein the harness also augments the issue identification with information descriptive of the issue identified.

33. A system in accordance with claim 32 wherein the analyzer programs also have provision to generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

34. A system in accordance with claim 32 wherein templates provide the information descriptive of the issues.

35. A system in accordance with claim 34 wherein the analyzer programs also have provision to generate, in addition to issue identification information, other issue-related information, including, wherein at least some of the templates contain one or more spaceholders for issue-related information, and wherein the harness inserts such issue-related information generated by an analyzer program into the template for an issue at positions indicated by the spaceholders.

36. A system in accordance with claim 21 wherein an output routine formulates the issue identification information generated by the analysis programs as an XML or similar document.

37. A system in accordance with claim 36 wherein the harness also augments the issue identification information with information descriptive of the issue identified.

38. A system in accordance with claim 37 wherein the analyzer programs also have provision to generate at least some of the information descriptive of the issue identified, tailoring this information in accordance with the specifics of each occurrence of each issue.

39. A system in accordance with claim 37 wherein templates provide the information descriptive of the issues.

40. A system in accordance with claim 39 wherein the analyzer programs also have provision to generate, in addition to issue identification information, other issue-related information, including, wherein at least some of the templates contain one or more spaceholders for issue-related information, and wherein the harness inserts such issue-related information generated by an analyzer program into the template for an issue at positions indicated by the spaceholders.

41. A system for generating information descriptive of issues arising in a monitored set of nodes comprising:

at least one analyzer means for accepting node-specific information, analyzing the information for the presence of issues, and generating issue identification information when an issue is detected;

harness means for exercising one or more analyzer means against node-specific information gathered from one or more nodes, presenting each analyzer means with at least some data from one of the nodes each time the analyzer means is exercised, and collecting any issue identification information generated by the analyzer means, said harness means also including means for augmenting the issue identification information generated with information identifying the node from which the node-specific information was obtained; and

an issues database containing the collected and augmented information for all the issues detected during such an exercising of the analyzer means against node-specific information.

* * * * *

US 20040006546A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0006546 A1

Wedlake et al. (43) Pub. Date: Jan. 8, 2004

(54) PROCESS FOR GATHERING EXPERT KNOWLEDGE AND AUTOMATING IT

(76) Inventors: William P. Wedlake, Suwanee, GA (US); Michael Joseph Brandon, Marietta, GA (US); Peter Van Giel, Mesnil-Saint-Blaise (BE); Jean Giuannotte, Grez-Doiceau (BE); Peter Put, Alberta (CA)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: 10/612,189

(22) Filed: Jul. 2, 2003

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/851,963, filed on May 10, 2001.

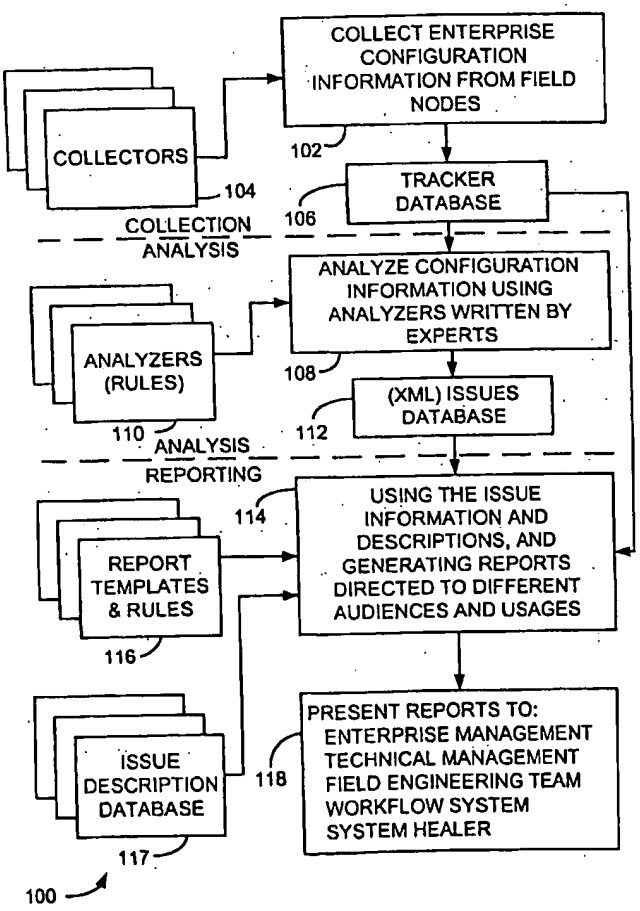
Publication Classification

(51) Int. Cl.[7] G06F 15/173; G06F 17/00; G06N 5/02; H03K 19/003; H05K 10/00; H02H 3/05; H04B 1/74; H04L 1/22; G06F 17/60

(52) U.S. Cl. 706/46; 705/1; 714/37; 714/38; 709/224

(57) ABSTRACT

A method is provided for auditing the configuration of an enterprise comprising the steps of: collecting information relating to the configuration of the enterprise, analyzing the configuration information based on expert knowledge; and providing the result of the analysis in the form of reports, and other results of the analyses.



What is claimed is:

1. A set of one or more analyzers to govern the analysis of information gathered from nodes or groups of nodes as well as the selective production of issue information, the analyzers comprising:

one or more analyzers which govern the analysis of specific configuration information to determine whether the software and/or the hardware associated with a node and/or a group of nodes is configured properly;

one or more analyzers which govern the analysis of specific information resulting from testing to determine whether the tested software and/or hardware associated with a node and/or a group of nodes is functioning properly;

one or more analyzers which govern the analysis of specific diagnostic and/or log information to determine whether the software and/or the hardware associated with a node and/or a group of nodes is functioning properly; and

these analyzers also govern the determination of whether and, if so, of what specific issue information is to be produced.

2. A set of one or more analyzers in accordance with claim 1, further comprising:

a harness or framework which, when provided with a list of one or more of the analyzers and with a list of one or more nodes or groups of nodes, exercises one or more of the analyzers against specific data associated with each node or group of nodes, captures any issue information produced, and augments such issue information with node identification information to produce an issues database.

3. A set of one or more analyzers in accordance with claim 1 wherein at least some analyzers include a code portion and a descriptor portion, the descriptor portion containing information identifying at least some of the specific node information required for the performance of the analysis defined by the code portion.

4. A set of one or more analyzers in accordance with claim 3 wherein at least some analyzers further include a template portion, the template portion defining the format of at least some specific issue information that may be generated as a result of the performance of the analysis defined by the code portion.

5. A set of one or more analyzers in accordance with claim 1 wherein at least some analyzers include a code portion and a template portion, the template portion defining the format of at least some specific issue information that may be generated as a result of the performance of the analysis defined by the code portion.

6. A set of one or more analyzers in accordance with claim 1 wherein the analyzers which govern the analysis of specific configuration information include:

at least one analyzer that analyzes the primary and secondary boot disks of one or more nodes, insuring that the boot disks are not installed on the same path and that their configurations are logical.

7. A set of one or more analyzers in accordance with claim 1 wherein the analyzers which govern the analysis of specific configuration information include:

at least one analyzer that analyzes the standard and backup kernel files of one or more nodes, insuring that the kernel files are not installed on the same path and that their configurations are logical.

8. A set of one or more analyzers in accordance with claim 1 wherein the analyzers which govern the analysis of specific configuration information include:

at least one analyzer that analyzes the configuration of service guard or equivalent software which software performs functional analyses of at least one mission critical switch guard or equivalent node cluster and which software switches one or more tasks from node to node within such a cluster to keep mission critical tasks running.

9. A set of one or more analyzers in accordance with claim 1 wherein the analyzers which govern the analysis of specific configuration information include:

at least one analyzer that analyzes the configuration of one or more hardware devices associated with a node or group of nodes.

10. A set of one or more analyzers in accordance with claim 1 wherein the analyzers which govern the analysis of specific configuration information include:

at least one analyzer that analyzes the system dump configurations of one or more nodes to see that the nodes are configured correctly to enable the performance of a successful system dump in case of a system crash or other failure.

11. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers which govern the analysis of specific diagnostic and/or log information include:

at least one analyzer that analyzes memory error indication information from one or more nodes to determine if memory error conditions are present and, if so, are serious enough to require attention.

12. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers which govern the analysis of specific diagnostic and/or log information include:

at least one analyzer that analyzes at least some data retrieved from a circular buffer containing diagnostic information and/or error messages.

13. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers which govern the analysis of specific diagnostic and/or log information include:

at least one analyzer that analyzes system logging file entries.

14. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers which govern the analysis of specific diagnostic and/or log information include:

at least one analyzer that analyzes an error log of I/O errors.

15. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers which govern the analysis of specific diagnostic and/or log information include:

at least one analyzer that examines and analyzes information extracted from CPU hardware logs of each processor installed upon a node.

16. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which analyzes disk usage data to determine the likelihood of "disk full" conditions occurring soon.

17. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which analyzes data indicating whether the service guard related daemons or their equivalents are operating properly to insure the continuance of mission critical tasks.

18. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which analyzes the operational status of at least one service guard or equivalent cluster of nodes, signaling an issue if such a cluster is down, or if one or more tasks of such a cluster is not running, or if no alternative node is available to which one or more tasks may be switched.

19. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which analyzes the status of one or more file systems.

20. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which data relating to the CPUs to determine if any have been de-configured.

21. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which analyzes the status of the CPU fans.

22. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which analyzes the status of the CPU power supplies.

23. A set of one or more analyzers in accordance with claim 1 wherein the one or more analyzers that govern the analysis of specific information resulting from testing includes:

at least one analyzer which analyzes the output generated by i/o device scans.

24. A method of analyzing information gathered from nodes or groups of nodes and selectively producing issue information, the method comprising:

analyzing one or more specific sets of configuration information to determine whether the software and/or the hardware associated with a node and/or a group of nodes is configured properly;

analyzing one or more specific sets of information resulting from testing to determine whether the tested software and/or hardware associated with a node and/or a group of nodes is functioning properly;

analyzing one or more sets of specific diagnostic and/or log information to determine whether the software and/or the hardware associated with a node and/or a group of nodes is functioning properly; and

determining whether and, if so, what specific issue information is to be produced as a result of these analyses, and producing that issue information.

25. A method of analyzing in accordance with claim 24, further comprising:

providing a list of one or more analyses and a list of one or more nodes or groups of nodes;

performing one or more of the listed analyses against specific data gathered from each listed node or group of nodes; and

capturing any issue information produced and augmenting such issue information with node identification information to produce an issues database.

26. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of configuration information includes:

analyzing the primary and secondary boot disks of one or more nodes to insure that the boot disks are not installed on the same path and that their configurations are logical.

27. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of configuration information includes:

analyzing the standard and backup kernel files of one or more nodes to insure that the kernel files are not installed on the same path and that their configurations are logical.

28. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of configuration information includes:

analyzing the configuration of service guard or equivalent software which software performs functional analyses of at least one mission critical switch guard or equivalent node cluster and which software switches one or more tasks from node to node within such a cluster to keep mission critical tasks running.

29. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of configuration information additionally comprises:

analyzing the configuration of one or more hardware devices associated with a node or group of nodes.

30. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of configuration information additionally comprises:

analyzing the system dump configurations of one or more nodes to see that the nodes are configured correctly to enable the performance of a successful system dump in case of a system crash or other failure.

31. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more sets of specific diagnostic and/or log information aditionally comprises:

analyzing memory error indication information from one or more nodes to determine if memory error conditions are present and, if so, are serious enough to require attention.

32. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more sets of specific diagnostic and/or log information additionally comprises:

analyzing at least some data retrieved from a circular buffer containing diagnostic information and/or error messages.

33. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more sets of specific diagnostic and/or log information additionally comprises:

analyzing system logging file entries.

34. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more sets of specific diagnostic and/or log information additionally comprises:

analyzing an error log of I/O errors.

35. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more sets of specific diagnostic and/or log information comprises:

analyzing information extracted from CPU hardware logs of each processor installed upon a node.

36. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyzing disk usage data to determine the likelihood of "disk full" conditions occurring soon.

37. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyzing data indicating whether the service guard related daemons or their equivalents are operating properly to insure the continuance of mission critical tasks.

38. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyzing the operational status of at least one service guard or equivalent cluster of nodes, signaling an issue if such a cluster is down, or if one or more tasks of such a cluster is not running, or if no alternative node is available to which one or more tasks may be switched.

39. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyzing the status of one or more file systems.

40. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyzing data relating to the CPUs to determine if any have been de-configured.

41. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyzing the status of the CPU fans.

42. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyze the status of the CPU power supplies.

43. A method of analyzing information in accordance with claim 24 wherein the step of analyzing one or more specific sets of information resulting from testing additionally comprises:

analyzing the output generated by i/o device scans.

44. A set of one or more analyzers to govern the analysis of information gathered from nodes or groups of nodes as well as the selective production of issue information, the analyzers comprising:

one or more configuration information analyzer means for analyzing specific configuration information to determine whether the software and/or the hardware associated with a node and/or a group of nodes is configured properly;

one or more test information analyzer means for analyzing specific information resulting from testing to determine whether the tested software and/or hardware associated with a node and/or a group of nodes is functioning properly;

one or more diagnostic and/or log information analyzer means for analyzing specific diagnostic and/or log information to determine whether the software and/or the hardware associated with a node and/or a group of nodes is functioning properly; and

issue information production means for determining whether and, if so, for determining what specific issue information is to be produced.

* * * * *

US 20030154177A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0154177 A1**

Holland et al. (43) **Pub. Date: Aug. 14, 2003**

(54) **COMBINED ISSUE IDENTIFYING ANALYZER AND INTELLIGENT REPORT GENERATOR**

(76) Inventors: **Paul Edward Holland**, Fort Collins, CO (US); **Adam Michael Carr**, Fort Collins, CO (US); **Mark William McDowell**, Fort Collins, CO (US)

Correspondence Address:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: **10/367,048**

(22) Filed: **Feb. 14, 2003**

Related U.S. Application Data

(63) Continuation-in-part of application No. 09/851,963, filed on May 10, 2001.

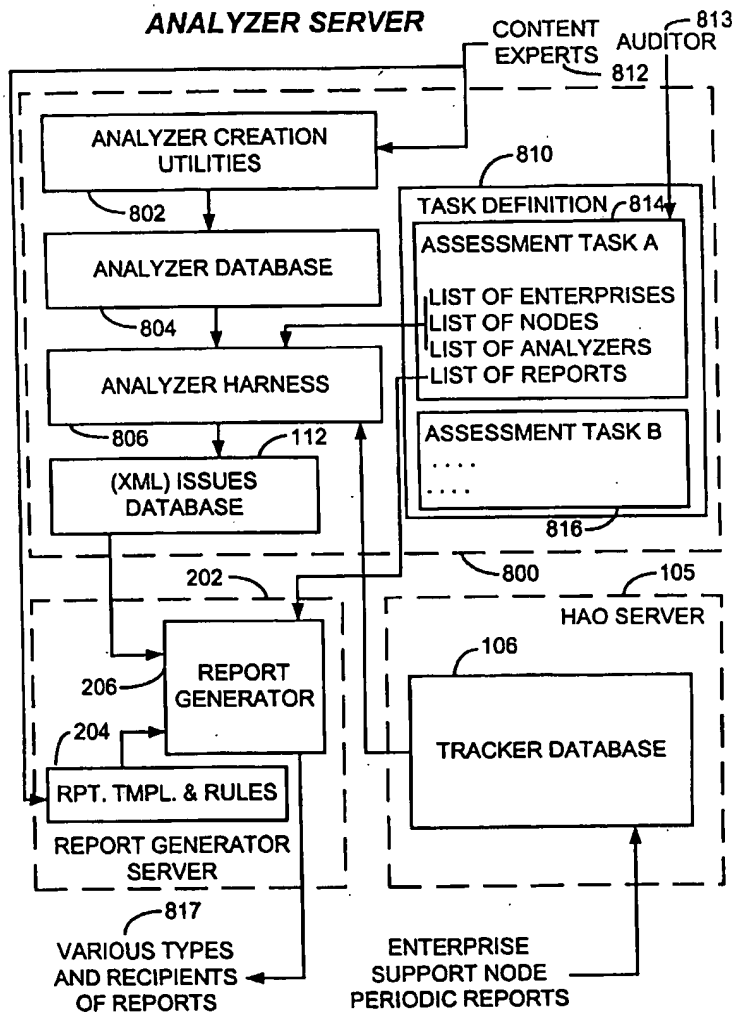
Publication Classification

(51) **Int. Cl.**[7] **G06F 17/00**; G06F 9/00; G06F 15/00

(52) **U.S. Cl.** **706/60**; 709/106; 715/500

(57) **ABSTRACT**

A method is provided for auditing the configuration of an enterprise comprising the steps of: collecting information relating to the configuration of the enterprise, analyzing the configuration information based on expert knowledge; and providing the result of the analysis in the form of reports, and other results of the analyses.



What is claimed is:

1. A method of generating a variety of reports descriptive of issues arising in monitored nodes of an enterprise or group of nodes, said method comprising the steps of:

analyzing node-specific information derived from one or more nodes, guided by rules of analysis, to detect the presence of one or more issues;

presenting at least some issue identification information together with node identification information;

providing several report template and/or rule sets each set defining the organization and contents of a class of reports; and

using two or more of these report templates and/or rule sets as a guide, and also guided by enterprise, node group, and/or node information that includes at least some of said issue and node identification information, generating two or more reports that present the information arranged, summarized, and/or supplemented in ways that suit the specialized needs of different audiences.

2. A method in accordance with claim 1 which includes the steps of providing at least one master report template and/or rule set directed to a particular audience and business need that includes varying combinations of subsidiary reports defined by other report templates and/or rule sets as well as introductory and explanatory information, and under its guidance generating at least one such master report.

3. A method in accordance with claim 1 wherein the method includes the steps of

providing at least one management summary report template and/or rule set, and

under its guidance generating one or more management summary reports that indicate, in a readable format, the general state of the enterprise or group of nodes.

4. A method in accordance with claim 1 wherein the method includes the steps of

providing at least one technical management summary report template and/or rule set, and

under its guidance generating one or more technical reports that list critical points in summary form.

5. A method in accordance with claim 4 wherein the method includes the step of providing within at least one such technical management summary report, in addition to the list of critical points, statements of their degrees of criticality.

6. A method in accordance with claim 4 wherein the method includes the step of providing within at least one such technical management summary report, in addition to the list of critical points, cost estimates for their resolution.

7. A method in accordance with claim 1 wherein the method includes the steps of

providing at least one technical detail report template and/or rule set, and

under its guidance generating one or more technical detail reports that state in detailed form at least some the specific issues for each of the nodes or group of nodes.

8. A method in accordance with claim 7 wherein the method includes the step of providing within at least one such technical detail report at least some snippets of node-specific information as needed to clarify an issue.

9. A method in accordance with claim 7 wherein the method includes the step of providing within at least one such technical detail report the motivation and/or the costs to address at least some issues.

10. A method in accordance with claim 7 wherein the method includes the step of providing within at least one such technical detail report details relating to at least some issues.

11. A system that generates a variety of reports descriptive of issues arising in the monitored nodes of an enterprise or group of nodes, said system comprising:

several analyzers which, when presented with data derived from a node, and when exercised under the guidance of rules of analysis, analyze the node data and generate output data signaling the presence of one or more issues indicated by the analysis;

a harness which exercises at least some the analyzers, providing them during each exercise with data gathered from a different node or group of nodes, and that presents issue data generated by an analyzer together with data identifying the node or group of nodes where the issue originated, to thereby generate an issues database;

report templates and/or rule sets each defining the organization and contents of a class of reports; and

a report generator guided by two or more report templates and/or rule sets and by data retrieved from the issues database through the process of generating two or more reports that present the information arranged, summarized, and/or supplemented in ways that suit the specialized needs of different audiences.

12. A system in accordance with claim 11 which further includes at least one master report template and/or rule set directed to a particular audience and business need that includes varying combinations of subsidiary reports defined by other templates and/or rule sets as well as introductory and explanatory information.

13. A system in accordance with claim 11 which further includes at least one management summary report template and/or rule set designed to be readable and to present the general state of the enterprise or group of nodes.

14. A system in accordance with claim 11 which further includes at least one technical management summary report template and/or rule set designed to list critical points in summary form.

15. A system in accordance with claim 14 in which at least one such technical management summary report, in addition to the list of critical points, contains statements of their degrees of criticality.

16. A system in accordance with claim 14 in which at least one such technical management summary report, in addition to the list of critical points, contains cost estimates.

17. A system in accordance with claim 11 which further includes at least one technical detail report template and/or rule set containing one or more technical detail reports capable of stating in detailed form at least some of the specific issues that may arise at each of the different nodes or group of nodes.

18. A system in accordance with claim 17 in which at least one such technical detail report contains at least some snippets of node-specific information as needed to clarify an issue that may arise.

19. A system in accordance with claim 17 in which at least one such technical detail report contains the motivations and/or the costs to address at least some of the issues that may arise.

20. A system in accordance with claim 17 in which at least one such technical detail report contains the details of at least some issues that may arise.

21. A system that generates a variety of reports descriptive of issues arising in the monitored nodes of an enterprise or group of nodes, said system comprising:

several analyzer means, designed to accept data derived from a node, for analyzing the node data, and for generating output data signaling the presence of one or more issues indicated by the analysis;

harness means for exercising at least some the analyzers, for providing them during each exercise with data gathered from a different node or group of nodes, and for presenting issue data generated by an analyzer together with data identifying the node or group of nodes where the issue originated, thereby generating an issues database;

report template and/or rule set means for defining the organization and contents of a class of reports; and

report generator means guided by two or more report templates and/or rule sets and by data retrieved from the issues database for generating two or more reports that present the information arranged, summarized, and/or supplemented in ways that suit the specialized needs of different audiences.

* * * * *

US 20030177106A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0177106 A1**

Brandon et al. (43) **Pub. Date: Sep. 18, 2003**

(54) **AUTOMATING EXPERT KNOWLEDGE WITH ANALYZER HARNESS**

(76) Inventors: **Michael Joseph Brandon**, Marietta, GA (US); **Adam M. Carr**, Fort Collins, CO (US); **Paul Edward Holland**, Fort Collins, CO (US); **Mark W. McDowell**, Fort Collins, CO (US)

Correspondence Address:
John C. Moran
HEWLETT-PACKARD COMPANY
Intellectual Propery Administration
P.O. Box 272400
Fort Collins, CO 80527-2400 (US)

(21) Appl. No.: **10/372,475**

(22) Filed: **Feb. 21, 2003**

Related U.S. Application Data

(63) Continuation of application No. 09/851,963, filed on May 10, 2001.

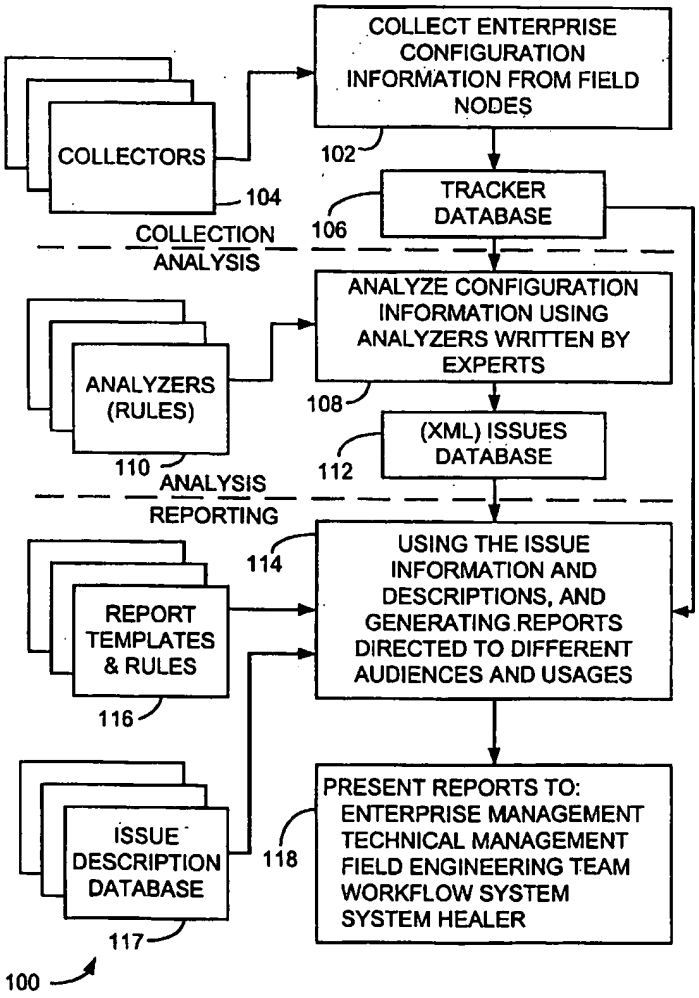
Publication Classification

(51) **Int. Cl.**[7] **G06F 17/00**

(52) **U.S. Cl.** **706/45**

(57) **ABSTRACT**

A method is provided for auditing the configuration of an enterprise comprising the steps of: collecting information relating to the configuration of the enterprise, analyzing the configuration information based on expert knowledge; and providing the result of the analysis in the form of reports, and other results of the analyses.



What is claimed is:

1. A system that analyzes data gathered from plural nodes, the system comprising:

a set of two or more analyzers designed by technical experts, each analyzer defining how to analyze specific information to determine whether and what analysis result information is to be produced; and

a harness accepting as input a list of analyzers and a list of nodes to be analyzed, the harness comprising

an analyzer loader that proceeds through the lists of analyzers and nodes, loading at least some of the analyzers to govern the analysis of each of the nodes,

an argument manager called upon by the analyzer loader during each analysis to gather the specific information from the node being analyzed for presentation to the analysis process, and

an output routine associated with the analyzer loader that receives any analysis result information produced and places it into at least one of a database or an XML (or equivalent) record, associating such information with the identity of the node being analyzed.

2. A system in accordance with claim 1 wherein the system is designed to analyze plural nodes on one or more enterprises, and wherein the list of nodes to be analyzed can be identified to the harness by the names of one or more enterprises some or all of whose nodes are to be analyzed.

3. A system in accordance with claim 2 wherein the harness is designed to proceed from one enterprise to the next; and

when analyzing one enterprise, the analyzer loader loads the analyzers in sequence and then exercises each analyzer repeatedly one or more times to analyze the one or more nodes of that enterprise which are to be analyzed by that analyzer.

4. A system in accordance with claim 1 wherein the analyzer loader loads the analyzers in sequence and then exercises each analyzer repeatedly one or more times to analyze the one or more nodes which are to be analyzed by that analyzer.

5. A system in accordance with claim 1 wherein the output routine formats result information as at least one XML (or equivalent) record which record is then expanded to include other information, including node identification information, such that an XML (or equivalent) report containing any analysis result information is generated.

6. A system in accordance with claim 5 wherein an analyzer issue template, associated with at least some analyzers, is incorporated into the XML (or equivalent) records that form the XML (or equivalent) report.

7. A system in accordance with claim 1 which further comprises:

a descriptor for each analyzer describing the specific information analyzed by the analyzer; and

wherein the argument manager is guided by these descriptors to gather the correct specific information from the nodes during the analysis process.

8. A system in accordance with claim 7 wherein:

the descriptor designates for each analyzer in which of several possible programming languages the analyzer is written; and

the analyzer loader varies the specific way in which it carries out the analysis process in accordance with the programming language of each analyzer.

9. A system in accordance with claim 1 which further comprises:

a report generator including report template or rule set or both which, guided by the analyses result information generated, generate a variety of reports for varying audiences.

10. A system in accordance with claim 1 which further comprises

a tracker database containing a historical record of specific information gathered from specific nodes at specific times or on specific dates or both;

the harness also accepts as input an indication of the specific times at which or the specific dates upon which (or both) the node information to be analyzed was gathered;

the argument manager gathers the specific information from records gathered at those specifric times or on those specific dates or both; and

the output routine also associates such analysis result information with the times at which or dates upon which (or both) the information was gathered.

11. A method for analyzing data gathered from plural nodes, the method comprising the steps of:

creating two or more analyzers, each analyzer defining how to analyze specific information to determine whether and what analysis result information is to be produced by the analyses;

guided by lists of analyzers and of nodes to be analyzed, performing at least some of the analyses defined by the analyzers with respect to each of the nodes;

during each analysis, gathering from the node under analysis the specific information required for the analysis; and

receiving any analysis result information produced during the analyses and placing it into at least one of a database or an XML (or equivalent) record, and associating this information with the identity of the node which is under analysis.

12. A method in accordance with claim 11 which includes the step of identiying the list of nodes to be analyzed by reference to the names of one or more enterprises some or all of whose nodes are to be analyzed.

13. A method in accordance with claim 12 wherein the analysis proceeds first from one enterprise to the next; and secondly, when proceding with the nodes of an enterprise, it proceeds from one analyzer to the next in sequence and performs each analysis repeatedly one or more times to analyze the one or more nodes of that enterprise which are to be analyzed by that analyzer.

14. A method in accordance with claim 11 wherein the analysis proceeds from one analyzer to the next in sequence

and performs each analysis repeatedly one or more times to analyze the one or more nodes which are to be analyzed by that analyzer.

15. A method in accordance with claim 11 which includes the step of formatting the result information as at least one XML (or equivalent) record, and then expanding that record to include other information, including node identification information, such that an XML (or equivalent) record containing any analysis result information is generated.

16. A method in accordance with claim 15 wherein the creation of at least some analyzers includes the creation of at least one analyzer issue template which is incorporated into the XML (or equivalent) records that form the XML (or equivalent) record.

17. A system in accordance with claim 11 wherein the creation of at least some analyzers includes the creation of a descriptor identifying the specific information to be analyzed by the analyzer; and

wherein these descriptors guide the gathering step.

18. A system in accordance with claim 17 wherein the creation step includes placing into at least some descriptors a designation of which of several possible programming languages the analyzer's analysis is written in; and

wherein the analysis process is varied in accordance with the programming language selected for each analyzer.

19. A system in accordance with claim 11 which further comprises the step of:

creating a variety of report template or rule sets or both; and

guided guided by these sets and the analyses result information record, generating a variety of reports.

20. A system in accordance with claim 11 which further comprises the steps of

providing a historical record of specific information gathered from specific nodes at specific times or on specific dates or both;

during each analysis, gathering from the node under analysis the specific information for specified times or specific dates (or both) or ranges thereof at which time or upon which dates (or both) the specific node information to be analyzed was gathered; and

associating any analysis result information with the times at which or the dates upon which (or both) the specific information was gathered.

21. A system that analyzes data gathered from plural nodes, the system comprising:

analyzer means for defining how specific information is to be analyzed to determine whether and what analysis result information is to be produced; and

a harness accepting as input a list of analyzers and a list of nodes to be analyzed, the harness comprising

analyzer loader means for proceeds through the lists of analyzers and nodes, causing at least some of the analyzers to govern the analysis of each of the nodes,

argument manager means called upon by the analyzer loader during each analysis for gathering the specific information from the node being analyzed for presentation to the analysis process, and

output routine means associated with the analyzer loader for receiving any analysis result information produced and for placing it into at least one of a database or an XML (or equivalent) record, associating such information with the identity of the node being analyzed.

* * * * *